TECH. MEMO
AW 121

UNLIMITED

TECH. MEMO
AW 121

DTIC FILE COPY

ROYAL AEROSPACE ESTABLISHMENT

AD-A215 866

TRACKING IN UNCERTAIN ENVIRONMENTS

by

D. J. Salmond

September 1989

Procurement Executive, Ministry of Defence
Farnborough, Hampshire

UNLIMITED

89 12 08 043

CONDITIONS OF RELEASE

0056199 BR-112043

MR PAUL A ROBEY
DTIC
Attn:DTIC-FDAC
Cameron Station-Bldg 5
Alexandria
VA 22304 6145

************************** U

************************** Y

Reports quoted are not necessarily available to members of the public or to commercial organisations.

UNLIMITED

ROYAL AEROSPACE ESTABLISHMENT

Technical Memorandum AW 121

Received for printing 4 September 1989

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TRACKING IN UNCERTAIN ENVIRONMENTS*

by

D. J. Salmond

SUMMARY

This study concerns the problem of tracking a target when the origin of the sensor measurements is uncertain. The full Bayesian solution to this type of problem gives rise to Gaussian mixture distributions, which are composed of an ever increasing number of components. To implement such a tracking filter, this growth of components must be controlled by approximating the mixture distribution.

Two algorithms have been developed for approximating Gaussian mixture distributions. These techniques attempt to minimize the number of mixture components without modifying the 'structure' of the distribution beyond a specified limit. Also the final approximation is itself a Gaussian mixture.

The performance of the algorithms has been assessed by simulation for the problem of tracking a single target in the presence of uniformly distributed false measurements. This assessment indicates the significant range of problem parameters where the new algorithms give a substantial performance improvement over the well known Probabilistic Data Association Filter (which approximates the mixture by a single Gaussian component).

The tracking example is extended in the second part of this study to show how the Bayesian approach may be applied to more complex uncertain tracking problems, including that of fusing data from several independent sources. In particular a computationally efficient filter is derived which improves the track estimate from a primary sensor, by making sub-optimal use of measurements from an auxiliary sensor. Finally, a general solution is derived for a tracking problem with multiple measurement classes. This general solution is used to derive a filter for tracking a target in the presence of intermittent interfering measurements, in addition to uniformly distributed false measurements.

* *This Technical Memorandum is taken from a thesis for the degree of D Phil of the University of Sussex.*

UNLIMITED

LIST OF CONTENTS

LIST OF CONTENTS (continued)

LIST OF CONTENTS (concluded)

1 INTRODUCTION

1.1 Background

A tracking filter is an algorithm for estimating the state (such as position and velocity) of an object from measurements of a sensor such as a radar. Following the usual convention, an object being tracked will be called a target. A basic assumption of most tracking filters, such as the α-β filter and other filters derived from Kalman filter theory, is that only measurements from the target of interest are passed to the filter. However in practice, sensors produce measurements as a result of random noise, clutter, interference and other targets, in addition to those from the required target. Usually it is not possible to distinguish with certainty between the wanted and the unwanted measurements. Hence there is a need for tracking filters which recognize that some of the received measurements may not originate from the required target.

Measurement origin uncertainty is most commonly encountered in the context of muliple target tracking, although in this study we shall only be concerned with the single target case. A number of approaches to the uncertain tracking problem, with the emphasis on multiple target tracking, are reviewed in the recent books by Blackman[1] and Bar-Shalom and Fortmann[2], and the survey papers[3-5]. There are essentially two types of approach to estimation in the presence of uncertainty: the decision-directed approach where decisions are taken and assumed to be true, and Bayesian techniques which allow for the possibility that the most likely option may be incorrect.

The simplest decision-directed technique is the 'nearest neighbour' filter: the track is updated with the measurement which is in some sense

closest to the expected target position. This is likely to give a poor result if several measurements occur in the vicinity of the expected target position. In these circumstances a branching or track splitting filter offers an improvement: a separate branch is propagated for each possible measurement. The growth of tracks is controlled by merging similar branches or by deleting branches if the likelihood function (or the support) of that branch falls below a certain threshold (see Smith and Buechler[6]). A more sophisticated approach is to choose the most likely hypothesis from the set of feasible hypotheses on the association of all measurements that have been received. This is a batch processing task (see Morefield[7]) which should provide an optimal solution in the maximum likelihood sense. Sequential versions of this method have also been derived. These are computationally convenient but sub-optimal (see Sittler[8], and Stein and Blackman[9]).

For this present study, the Bayesian approach has been adopted. As already indicated, this approach avoids the need to make 'hard' decisions among quite probable hypotheses. Also an obvious implementation is via a recursive filter which is convenient for real time processing. However the full Bayesian solution is impractical and some approximation is essential; promising results have been obtained by a number of authors[10-26]. Approximation of the optimal solution is one of the main subjects of this study.

An approximate Bayesian filter for the problem of tracking a a single target in clutter was first formulated by Singer *et al*[10]. For the same problem, a very efficient approximation technique known as the Probabilistic Data Association Filter (PDAF) was proposed by Bar-Shalom and Tse[11]. Various extensions of the basic PDAF for special cases including target maneouvres[12], random

measurement arrival times[13] and dual sensors[14] have been developed by Bar-Shalom and co-workers. An extension of the PDAF to the multiple target case was reported by Bar-Shalom[17] and Fortmann *et al*[18,19] (also see Refs 20 to 22). An important paper by Reid[23] presents a Bayesian multiple target filter which does not use the PDAF approximation. The branching algorithm of Smith and Buechler[6] may be viewed as a much simplified version of this filter. More recent work on Bayesian multiple target tracking is reported by Mori *et al*[26].

1.2 The Bayesian approach

In the Bayesian approach to tracking, one attempts to construct the probability density function (pdf) of the target state $\underline{x}$, based on all available information including the set Z of received measurements[27]. The required conditional pdf of $\underline{x}$ may be written $p(\underline{x}|Z)$. Since this pdf embodies all available statistical information, it may be said to be the complete solution of the tracking problem. In principle, an optimal estimate of $\underline{x}$ for any criterion may be obtained from $p(\underline{x}|Z)$. A measure of the accuracy of the estimate may also be derived from $p(\underline{x}|Z)$. Clearly it is most desirable to obtain this conditional pdf whenever an estimate of the target state is required.

For many tracking problems an estimate is required every time that a set of sensor measurements is received. In this case a recursive filter is a convenient solution. Such a filter consists of essentially two stages: prediction and update. For prediction it is assumed that an equation describing the evolution of the target state is available. This can be used to predict the pdf of state forwards from one measurement time to the next. Since the target is usually subject to unknown disturbances, prediction usually increases the covariance of the state pdf. The update operation uses the latest set of measurements to

modify the predicted pdf. This is conveniently achieved using Bayes theorem which is the mechanism for updating a pdf or probability in the light of extra information from new data.

For estimation problems where the origin of measurements is known, Bayes theorem leads to the Kalman filter update relations[28-32], provided that the problem is linear and all random elements are Gaussian (see Appendix A). This is the optimal tracking filter for this standard tracking problem, and in this case $p(\underline{x}|Z)$ is a Gaussian pdf. This is not so if the measurement origin is uncertain. To construct the required pdf in this case it is necessary to take account of all possible measurement associations. For each of these possibilities or hypotheses, there is a corresponding Gaussian pdf of target state. Thus the overall pdf of target state is a Gaussian mixture pdf[33,34] of the form:

$$p(\underline{x}|Z) = \sum_{i=1}^{N} \beta_i \, p_i(\underline{x}) \quad , \tag{1.1}$$

where $p_i(\underline{x})$ is the Gaussian pdf corresponding to hypothesis i, N is the total number of feasible hypotheses at this time and β_i is the probability that hypothesis i is correct, such that:

$$\beta_i > 0 \qquad \text{and} \qquad \sum_{i=1}^{N} \beta_i = 1 \ .$$

When new measurements are received for the update of this pdf, the number of feasible hypotheses from past measurements is compounded by origin uncertainty in the latest set. Since the probability and the pdf corresponding to each of these hypotheses have to be updated via Bayes theorem, it is clear that the computational requirements of the

full Bayesian solution increase rapidly as tracking proceeds. This is the major difficulty of the Bayesian approach to the uncertain tracking problem.

1.3 A practical sub-optimal filter

To implement a Bayesian filter it is essential to contain these computational requirements within acceptable bounds by making approximations. Any approximation which changes $p(\underline{x}|Z)$ renders the filter sub-optimal, so the aim should be to achieve the necessary reduction in computation with minimal performance penalty. At each measurement time, it is usual practice to subject the received measurements to a coarse acceptance test. This rejects any data that are very unlikely to originate from the target, so that very improbable hypotheses are not considered. However origin uncertainty amongst the accepted measurements may still cause the number of mixture components of equation (1.1) to grow rapidly. Thus further direct approximation of the mixture distribution may be necessary. Unlike the acceptance test this approximation may result in a significant modification of the complete solution, and so the choice of approximation should be carefully considered.

As already mentioned, the PDAF[11] is a popular and economical scheme for approximating the mixture. This method reduces the complete mixture to a single Gaussian component after processing each set of sensor measurements. However this may destroy valuable information, especially if several significant well spaced components are present. To provide a better approximation to the mixture, two new algorithms (the Clustering Algorithm and the Joining Algorithm) have been derived which allow more than one component to be retained. These mixture reduction algorithms operate by merging similar components together, and they

are based on the requirement that reduction should proceed with minimal modification to the 'structure' of the distribution (see Chapter 3).

1.4 The baseline problem and simulation studies

A baseline problem has been chosen for this study to provide a specific example of the growth in the number of measurement association hypotheses, and to show how the reduction algorithms may be applied to control this. The problem is to track a single target from sensor data which includes spurious as well as useful measurements. A set of sensor measurements is produced at discrete time intervals. Each set is composed of at most one true measurement which originates from the target, and a number of false measurements which are uniformly distributed over the measurement space and are independent of the target. The true measurement has a Gaussian distribution about the target position and it cannot be distinguished from the false measurements. The target moves according to a linear model driven by Gaussian noise. The full Bayesian solution to this problem, which has been considered by several authors[10,11], is derived in Chapter 2. All of the tracking examples considered in this study are variations of this single target problem. The single target case suffices to investigate the trade off between complexity and filter performance. Also the techniques developed here could be adapted for the multiple target problem.

For our purposes, the performance of the mixture reduction algorithms depends on the performance of the tracking filters that employ them. The primary measure of filter performance chosen for this study is the average time for which the filter maintains track on a target, ie the average track lifetime. Since there is no tractable

analytical means of evaluating this performance measure, Monte Carlo simulations have been carried out for a particular example of the baseline problem. In this example, which is also used by Bar-Shalom and Birmiwal[15], the target moves in a plane, the target kinematics are described by a second order model and sensor measurements consist of Cartesian co-ordinate pairs. The 'difficulty' of this tracking problem may be easily controlled by adjusting several problem parameters. In Chapter 4, the performance of tracking filters using the new reduction algorithms is examined in detail for a single set of problem parameters. In particular the effect of varying the maximum number of mixture components retained by the reduction algorithms is investigated. For this and other simulations in this study, the PDAF provides the performance reference against which other filters are compared. The results of Chapter 4 indicate that the Clustering Algorithm is more computationally efficient than the Joining Algorithm, and so from Chapter 5 onwards the former reduction technique is employed.

In Chapter 5, the performance of a filter using the Clustering Algorithm is compared with the PDAF over a wide range of problem parameters for the simulation example. The new filter should always outperform the PDAF, since the Clustering Algorithm retains more information. We have attempted to identify the approximate region of the problem parameter space where the performance of the Clustering Algorithm filter is *significantly* better than the PDAF, *ie* where it is worth retaining more than one component.

A second 'sector scan' example of the baseline problem is considered in Chapter 6. This example has been used to examine the effect of several practical filtering difficulties. These include sensor measurements in polar co-ordinates which are a non-linear function of the target state, and target manoeuvres which are not correctly represented by the filter's assumed target model.

As well as providing a tool for evaluating a performance measure, simulation is a useful aid to understanding the operation of a filter. For this study the simulation programs have been designed so that either multiple replications can be performed to generate performance statistics, or single runs can be carried out to examine filter operation in detail. For multiple runs, overall performance measures are produced together with a summary of the results of each individual replication, including its random number seeds. Thus any replication may be rerun with the program in single replication mode to produce detailed output files for a thorough analysis of filter operation. All simulation programs were written in Fortran 77, and use of the Cray 1S computer at RAE Farnborough enabled an extensive range of simulation experiments to be performed.

1.5 Extensions of the baseline problem

The final part of this study is concerned with extensions of the baseline problem. In Chapter 7 we consider the problem of fusing information from a number of sources. For many sensors it is possible to obtain information on the origin of a measurement by analysing the signal from which it is derived. For example, the shape of the return from a pulse radar or the fluctuation over several returns may indicate whether the measurement originates from clutter or from a true target. Clearly the filter should make use of this signature information, and Nagarajan *et al*[35] show how it may easily be included in the Bayesian formulation to modify $p(\underline{x}|Z)$. Also in many tracking systems, measurements are available from several independent sensors. Data from each of these sensors may be incorporated sequentially because they are independent. In Section 7.4 we consider the particular data fusion problem of combining

information from a primary sensor which produces range and bearing position measurements with an independent auxiliary sensor. The auxiliary sensor gives only bearing information but it does include an imperfect classification of each of its measurements. A new filter has been derived for this problem which uses the auxiliary measurements in a sub-optimal but efficient way. The performance of the filter is compared with the single sensor filter to show the value of sub-optimal processing of the auxiliary measurements.

For Chapters 2 to 7 it is assumed that a measurement from a given sensor is a sample from one of two distributions: true or false. In Chapter 8 we extend this to allow for samples from more than two distributions, *ie* more than two classes of measurement are allowed. The general solution to this problem is derived. This general solution is used to develop a practical filter for tracking a target in the presence of intermittent interference, in addition to the usual false measurements.

2 THE BASELINE PROBLEM: TRACKING A SINGLE TARGET IN THE PRESENCE OF RANDOM UNIFORMLY DISTRIBUTED FALSE MEASUREMENTS

2.1 Introduction

In this chapter a formal statement of the baseline problem is given and the optimal Bayesian solution of this problem is derived. This problem, which is taken from Refs 10 and 11, provides a convenient example which illustrates many of the difficulties of uncertain tracking, and it is a suitable basis for extension to more complex problems.

A full account of the solution is presented here to facilitate the description of extensions given in later chapters. The major result is that the posterior pdf of target state at each time step is a Gaussian mixture and that the number of components which comprise this mixture increases with time. This is confirmed by induction. Assuming that the prior pdf of the target state at time step k is a Gaussian mixture, the posterior distribution, after updating with the measurements received at this time step, is shown to be another Gaussian mixture with an increased number of components (sections 2.3.1 and 2.3.2). Using the target model, this posterior pdf is projected forwards to show that the prior pdf at the following time step $k+1$ is also a Gaussian mixture (section 2.3.3), so completing the proof. The recurrence relations for updating and prediction are given, and the solution is seen to be equivalent to a bank of parallel Kalman filters whose number grows with time. The significance of an optimal solution requiring propagation of an ever increasing number of Gaussian components is discussed in section 2.4.

2.2 Problem statement

The problem is to provide an estimate of the state $\underline{x}$ of a single target at discrete time steps, based on all the available information. The state vector $\underline{x}$ typically consists of target position and velocity, but other attributes of the target may also be included. It is assumed that $\underline{x}$ evolves according to a linear recurrence relation of the form:

$$\underline{x}_{k+1} = \Phi \underline{x}_k + \Gamma \underline{w}_k \quad , \tag{2.1}$$

where $\underline{x}_k$ is the n-component state vector at time t_k ,

Φ is the $n \times n$ state transition matrix,

Γ is an $n \times r$ matrix

and $\underline{w}_k$ is an r-component vector of system driving noise which has a Gaussian distribution with zero mean and covariance given by:

$$E\left[\underline{w}_i \underline{w}_k^T\right] = Q\delta_{ik} \quad .$$

Here Q is a positive definite $r \times r$ matrix and δ_{ik} is the Kronecker delta. Equation (2.1) describes the kinematics of the target and is known as the target model. Initially, at time t_1 , the state vector $\underline{x}_1$ is assumed to have a Gaussian distribution with known mean $\bar{\underline{x}}_1$ and covariance M_1 (a positive definite $n \times n$ matrix).

At every time step k , a single sensor scans a surveillance region and passes a set Z_k of m_k measurements to the tracking filter:

$$Z_k = \left\{ \underline{z}_{kj} : \ j = 1 , \ldots , m_k \right\} .$$

Each measurement $\underline{z}_{kj}$ is a u-component vector. It is assumed that the target is well inside the surveillance region of the sensor, but that the (known) probability P_D of detecting the target may be less than

unity. It is also assumed that at most one of the measurements may originate from the target. If measurement $\underline{z}_{kj}$ does originate from the target, then it is related to the state vector by the linear relationship:

$$\underline{z}_{kj} = H\underline{x}_k + \underline{v}_k \ , \qquad (2.2)$$

where H is the $u \times n$ measurement matrix

and $\underline{v}_k$ is a u-component vector of measurement noise which has a Gaussian distribution with zero mean and covariance given by:

$$E\left[\underline{v}_i \underline{v}_k^T\right] = R\delta_{ik} \ .$$

Here R is a positive definite $u \times u$ matrix and δ_{ik} is the Kronecker delta. A measurement which originates from the target is said to be true, while all other measurements are false. A false measurement is assumed to be independent of the state vector, to have a uniform distribution over the surveillance region of the sensor and to be independent of all other present and past measurements. False measurements are assumed to occur at an average density of ρ per unit area. Further it is assumed that before examining the values of the measurements in the set Z_k, there is no information on which, if any, of the measurements are associated with the target.

The following information is available to the tracking filter:

(i) The distribution of the initial state vector including its mean $\bar{\underline{x}}_1$ and covariance M_1 .

(ii) The target model, equation (2.1), including Φ and Γ .

(iii) The relationship between the state vector and the true measurement, equation (2.2), including H .

(iv) The statistics of the false measurements, the true measurement and the model driving noise, including ρ, R and Q.

(v) The detection probability P_D of the sensor.

(vi) The measurement sets Z_k for all past and current time steps.

The tracking filter does not know:

(i) The values of the state vector $\underline{x}_k$, or the noise vectors $\underline{v}_k$ and $\underline{w}_k$ at any time step.

(ii) The identity of the true measurement.

Note that if the identity of the true measurement were known, the problem would reduce to that of the standard Kalman filter (see Appendix A).

2.3 The Bayesian solution

2.3.1 The prior distribution of the state vector at time t_k

The prior pdf of the state vector at time t_k is the pdf of $\underline{x}_k$ given all available information up to time t_k but excluding the set of measurements received at time t_k. This available prior information at time t_k is denoted $\mathcal{P}_k$, and this includes all measurements received at the previous time steps:

$$Z_1 , Z_2 , \ldots , Z_{k-1} .$$

Since any one or none of the measurements of Z_i could be true, there are exactly $m_i + 1$ exclusive hypotheses concerning the truth or

falsehood of the members of Z_i . Thus the total number of possible hypotheses under $\mathscr{P}_k$ is:

$$n_{k-1} = \prod_{i=1}^{k-1}\left(m_i + 1\right) \quad . \tag{2.3}$$

Therefore, given n_{k-1} possible hypotheses, the pdf of the state vector $\underline{x}_k$ may be written:

$$p\left(\underline{x}_k \middle| \mathscr{P}_k\right) = \sum_{i=1}^{n_{k-1}} p\left(\underline{x}_k \middle| \mathscr{H}_{k-1\ i}, \mathscr{P}_k\right) \Pr\left\{\mathscr{H}_{k-1\ i} \middle| \mathscr{P}_k\right\} \quad . \tag{2.4}$$

Here $\mathscr{H}_{k-1\ i}$ denotes one of the possible hypotheses on the measurements available under $\mathscr{P}_k$, $p\left(\underline{x}_k \middle| \mathscr{H}_{k-1\ i}, \mathscr{P}_k\right)$ is the pdf of $\underline{x}_k$ assuming $\mathscr{H}_{k-1\ i}$ is correct and $\mathscr{P}_k$ is given, and $\Pr\left\{\mathscr{H}_{k-1\ i} \middle| \mathscr{P}_k\right\}$ is the probability that $\mathscr{H}_{k-1\ i}$ is correct given the information $\mathscr{P}_k$. In expression (2.4), the prior pdf of $\underline{x}_k$ is written as the weighted sum over all possible hypotheses of the pdf of $\underline{x}_k$ conditional on each hypothesis. The weighting factors in the summation are the corresponding prior probabilities of each hypothesis being true. Equation (2.4) is intuitively reasonable and is sometimes known as the total probability theorem.

Now suppose that the conditional pdfs in the RHS of equation (2.4) are known to be Gaussian, *ie*

$$p\left(\underline{x}_k \middle| \mathscr{H}_{k-1\ i}, \mathscr{P}_k\right) = \mathscr{N}\left(\underline{x}_k ; \bar{\underline{x}}_{ki}, M_{ki}\right) \quad , \tag{2.5}$$

where $\bar{\underline{x}}_{ki}$ and M_{ki} are known, and $\mathscr{N}(\underline{a}; \underline{b}, C)$ denotes a Gaussian pdf evaluated at $\underline{a}$ with mean $\underline{b}$ and covariance C. Also suppose that the probabilities of the hypotheses are known and are denoted:

$$\Pr\left\{\mathscr{H}_{k-1\ i} \middle| \mathscr{P}_k\right\} = \beta_{k-1\ i} \quad . \tag{2.6}$$

In this case equation (2.4) is a fully specified Gaussian mixture pdf where each Gaussian component corresponds to one of the possible hypotheses. Note that the above suppositions are true for $k = 1$, in which case, from the problem statement,

$$p\left(\underline{x}_1 \middle| \mathscr{P}_1\right) = \mathscr{N}\left(\underline{x}_1 \ ; \ \bar{\underline{x}}_1 \ , \ M_1\right) \quad ,$$

which is a degenerate Gaussian mixture with a single component.

2.3.2 The posterior pdf of the state vector

The set Z_k of m_k measurements received at time t_k is to be used to update the prior pdf of $\underline{x}_k$ specified by equations (2.4) to (2.6). The resulting posterior pdf is denoted:

$$p\left(\underline{x}_k \middle| Z_k \ , \ \mathscr{P}_k\right) \quad .$$

In the following working we shall omit $\mathscr{P}_k$ for ease of notation, although the dependency should be understood for all conditional probabilities and pdfs. Thus the posterior pdf of $\underline{x}_k$ will be written:

$$p\left(\underline{x}_k \middle| Z_k\right) \quad .$$

After updating with the latest set of measurements, the total number of possible hypotheses is increased to:

$$n_{k-1}\left(m_k + 1\right) \quad .$$

This increase may be viewed as a branching process where each of the $\mathscr{H}_{k-1\ i}$ prior hypotheses of equation (2.4) may be seen as a potential

track and each of these tracks then splits into a further $m_k + 1$ tracks resulting from the new set of measurements. Thus a posterior hypothesis including the latest set of measurements Z_k may be written as a joint hypothesis:

$$\mathscr{H}'_{kij} = \left(\mathscr{H}_{k-1\ i}\ ,\ \Psi_{kj}\right)$$

where Ψ_{kj} is independent of $\mathscr{H}_{k-1\ i}$ and indicates that the jth measurement of set Z_k is true (or that they are all false if $j = 0$). The complete set of posterior hypotheses is:

$$\left\{\mathscr{H}'_{kij} : i = 1, \ldots, n_{k-1} ; j = 0, \ldots, m_k\right\} .$$

Hence the posterior of pdf of $\underline{x}_k$ may be written in the form:

$$p\left(\underline{x}_k \middle| Z_k\right) = \sum_{i=1}^{n_{k-1}} \sum_{j=0}^{m_k} p\left(\underline{x}_k \middle| \mathscr{H}'_{kij}\ ,\ Z_k\right) \Pr\left\{\mathscr{H}'_{kij} \middle| Z_k\right\} . \tag{2.7}$$

First consider the posterior pdf of $\underline{x}_k$ conditioned by $\mathscr{H}'_{kij}$:

$$p\left(\underline{x}_k \middle| \mathscr{H}'_{kij}\ ,\ Z_k\right)$$

is the probability density resulting from updating $p\left(\underline{x} \middle| \mathscr{H}_{k-1\ i}\right)$ on the assumption that the jth measurement from Z_k is true (for $j \neq 0$). In this case $\underline{z}_{kj}$ is the only useful measurement from Z_k and the other members of Z_k can be discarded since they contain no relevant information. A true measurement $\underline{z}_T$ has a Gaussian distribution:

$$\mathscr{N}\left(\underline{z}_T\ ;\ H\underline{x}_k\ ,\ R\right) ,$$

and the prior density of $\underline{x}_k$ under $\mathscr{H}_{ki}$ is also Gaussian, given by equation (2.5). Hence the required posterior density is also Gaussian

and is given by the standard Kalman filter (see Appendix A).

So for $j \neq 0$:

$$
\left.
\begin{aligned}
p\left(\underline{x}_k \middle| \mathscr{H}'_{kij}, Z_k\right) &= \mathcal{N}\left(\underline{x}_k; \hat{\underline{x}}'_{kij}, P'_{kij}\right) \\
\text{where} \qquad \hat{\underline{x}}'_{kij} &= \bar{\underline{x}}_{ki} + K_{ki}\left(\underline{z}_{kj} - H\bar{\underline{x}}_{ki}\right) , \\
K_{ki} &= P'_{kij}H^T R^{-1} , \\
P'_{kij} &= M_{ki} - M_{ki}H^T S_{ki}^{-1} H M_{ki} \\
\text{and} \qquad S_{ki} &= HM_{ki}H^T + R .
\end{aligned}
\right\} \tag{2.8}
$$

If $j = 0$, none of the members of Z_k are true and so the prior pdf is not modified:

$$
\left.
\begin{aligned}
\hat{\underline{x}}'_{ki0} &= \bar{\underline{x}}_{ki} \\
\text{and} \qquad P'_{ki0} &= M_{ki} .
\end{aligned}
\right\} \tag{2.9}
$$

Now turning to the second term in the summation of equation (2.7), the posterior probability that $\mathscr{H}'_{kij}$ is correct may be evaluated using Bayes theorem:

$$
\Pr\left\{\mathscr{H}'_{kij} \middle| Z_k\right\} = \frac{p\left(Z_k \middle| \mathscr{H}'_{kij}\right)\Pr\left\{\Psi_{kj} \middle| \mathscr{H}_{k-1\ i}\right\}\Pr\left\{\mathscr{H}_{k-1\ i}\right\}}{p\left(Z_k\right)} , \tag{2.10}
$$

where $p(Z_k)$ is a normalizing constant given by:

$$
p\left(Z_k\right) = \sum_{i=1}^{n_{k-1}} \sum_{j=0}^{m_k} p\left(Z_k \middle| \mathscr{H}'_{kij}\right)\Pr\left\{\Psi_{kj} \middle| \mathscr{H}_{k-1\ i}\right\}\Pr\left\{\mathscr{H}_{k-1\ i}\right\} .
$$

The equation (2.10) indicates how the prior probability $\Pr\left\{\mathscr{H}_{k-1\ i}\right\}$ is modified by the observations at time t_k. The posterior probability can be found by evaluating the three factors in the numerator of the RHS of equation (2.10).

First consider $p\left(Z_k \middle| \mathscr{H}'_{kij}\right)$. This may be written:

$$p\left(Z_k \middle| \mathscr{H}'_{kij}\right) = \int p\left(Z_k , \underline{x}_k \middle| \mathscr{H}'_{kij}\right) \underline{dx}_k = \int p\left(Z_k \middle| \underline{x}_k , \mathscr{H}'_{kij}\right) p\left(\underline{x}_k \middle| \mathscr{H}'_{kij}\right) \underline{dx}_k \quad \ldots\ldots (2.11)$$

Since the elements of Z_k are independent:

$$p\left(Z_k \middle| \underline{x}_k , \mathscr{H}'_{kij}\right) = \prod_{\ell=1}^{m_k} p\left(\underline{z}_{k\ell} \middle| \underline{x}_k , \Psi_{kj}\right) .$$

A measurement $\underline{z}_{k\ell}$ is false under Ψ_{kj} if $j \neq \ell$. False measurements are uniformly distributed over the surveillance region of the sensor, and so the pdf of a false measurement is V_k^{-1}, where V_k is the volume of the surveillance region. If $j = \ell$, the measurement $\underline{z}_{k\ell}$ is true and so is a sample from the Gaussian distribution defined by equation (2.2). The prior pdf of $\underline{x}_k$:

$$p\left(\underline{x}_k \middle| \mathscr{H}'_{kij}\right) = p\left(\underline{x}_k \middle| \mathscr{H}_{k-1\ i}\right) ,$$

which is the Gaussian pdf (2.5). Hence on substituting into equation (2.11) we obtain, for $j \neq 0$:

$$\begin{aligned} p\left(Z_k \middle| \mathscr{H}'_{kij}\right) &= V_k^{-m_k+1} \int \mathcal{N}\left(\underline{z}_{kj} ; H\underline{x}_k , R\right) \mathcal{N}\left(\underline{x}_k ; \bar{\underline{x}}_{ki} , M_{ki}\right) \underline{dx}_k \\ &= V_k^{-m_k+1} \mathcal{N}\left(\underline{z}_{kj} ; H\bar{\underline{x}}_{ki} , S_{ki}\right) , \end{aligned} \qquad (2.12)$$

where S_{ki} is defined in the relations (2.8) and the integral is evaluated in Appendix A.2. Expression (2.12) is strictly correct only for a surveillance region of infinite extent. However, the truncation effect is negligible provided that, for each component of $\underline{z}_{kj}$, the distance from $H\underline{\bar{x}}_{ki}$ to the boundary of the surveillance region is large compared with the standard deviation of the component. If $j = 0$ so all the measurements are false:

$$p\left(Z_k \middle| \mathscr{H}'_{ki0}\right) = V_k^{-m_k} . \tag{2.13}$$

The second factor in the numerator of (2.10) is the prior probability of Ψ_{kj} :

$$\Pr\left\{\Psi_{kj} \middle| \mathscr{H}_{k-1\ i}\right\} = \Pr\left\{\Psi_{kj}\right\}$$

since the hypothesis on the current set of measurements is independent of hypotheses on measurements from previous time steps. The only prior information available is the probability P_D of detecting the target and the probability of the sensor receiving m false measurements. If false measurements are uniformly distributed over the measurement space with density ρ , then it can be shown that the probability of m false measurements falling within the surveillance region of the sensor is given by a Poisson distribution. If the volume of the surveillance region is V_k , the probability of receiving m false measurements is given by:

$$g(m) = e^{-\rho V_k}\left(\rho V_k\right)^m / m! . \tag{2.14}$$

The hypothesis Ψ_{k0} corresponds to the event of failing to detect the target and receiving m_k false measurements. The prior probability of this occurrence is:

$$\Pr\left\{\Psi_{k0}\right\} = \left(1 - P_D\right) g\left(m_k\right) . \tag{2.15}$$

Any of the hypotheses Ψ_{kj} , $j \neq 0$, could correspond to the situation of detecting the target and receiving $m_k - 1$ false measurements. *A priori*, each of these hypotheses is equally probable, and since there are m_k of them (for $j \neq 0$) :

$$\Pr\left\{\Psi_{kj}\right\} = P_D g\left(m_k - 1\right) \Big/ m_k . \tag{2.16}$$

The third factor in the numerator of (2.10) is given directly by equation (2.6):

$$\Pr\left\{\mathcal{H}_{k-1\ i}\right\} = \beta_{k-1\ i} . \tag{2.17}$$

Substituting (2.12) to (2.17) into (2.10) we obtain:

$$\Pr\left\{\mathcal{H}'_{kij} \middle| Z_k\right\} = \begin{cases} \dfrac{\beta_{k-1\ i} \mathcal{N}\left(\underline{z}_{kj} ; H\bar{\underline{x}}_{ki} , S_{ki}\right)}{E} & \text{for } j \neq 0 \\ \\ \dfrac{\beta_{k-1\ i}\left(1 - P_D\right)\rho}{P_D E} & \text{for } j = 0 \end{cases}$$

...... (2.18)

where $E = \dfrac{\left(1 - P_D\right)\rho}{P_D} + \displaystyle\sum_{r=1}^{n_{k-1}} \beta_{k-1\ r} \sum_{\ell=0}^{m_k} \mathcal{N}\left(\underline{z}_{k\ell} ; H\bar{\underline{x}}_{kr} , S_{kr}\right)$

is the normalizing denominator. This equation is of key importance because it defines the weightings of the mixture distribution (2.7). Note that the volume V_k of the surveillance region does not appear in (2.18). Also note that if $P_D = 1$, knowledge of the density ρ of false measurements does not contribute to the posterior pdf.

Thus the posterior pdf of $\underline{x}_k$ given by equation (2.7) is a fully specified Gaussian mixture. Equation (2.7) can be rewritten as a single sum by defining:

$$\mathscr{H}_{k\ell} = \mathscr{H}'_{kij}$$

$$\hat{\underline{x}}_{k\ell} = \hat{\underline{x}}'_{kij}$$

$$P_{k\ell} = P'_{kij}$$

and

$$\beta_{k\ell} = \Pr\left\{\mathscr{H}'_{kij} \middle| Z_k\right\}$$

where $\ell = (i - 1)\left(m_k + 1\right) + j + 1$, for $i = 1 , \ldots , n_{k-1}$ and $j = 0 , \ldots , m_k$. Thus:

$$p\left(\underline{x}_k \middle| Z_k\right) = \sum_{\ell=1}^{n_k} p\left(\underline{x}_k \middle| \mathscr{H}_{k\ell} , Z_k\right) \Pr\left\{\mathscr{H}_{k\ell} \middle| Z_k\right\} , \qquad (2.19)$$

where $n_k = n_{k-1}\left(m_k + 1\right) = \prod_{i=1}^{k} \left(m_i + 1\right)$,

$$p\left(\underline{x}_k \middle| \mathscr{H}_{k\ell} , Z_k\right) = \mathscr{N}\left(\underline{x}_k ; \hat{\underline{x}}_{k\ell} , P_{k\ell}\right)$$

and $\Pr\left\{\mathscr{H}_{k\ell} \middle| Z_k\right\} = \beta_{k\ell}$.

The Gaussian mixture (2.19) contains all the available information on the state vector $\underline{x}_k$ after taking account of the latest set of measurements Z_k . Thus in principle, the optimal estimate based on any desired criterion may be obtained from (2.19). This is considered in section 2.4.

2.3.3 The prior pdf of the state vector at time t_{k+1}

To establish, by induction, the general property that the prior pdf of $\underline{x}_k$ (equation (2.4)) is a fully specified Gaussian mixture, it is necessary to derive the pdf of $\underline{x}_{k+1}$ from the result (2.19). This pdf may be derived from $p\left(\underline{x}_k \middle| Z_k , \mathscr{P}_k\right)$ (note $\mathscr{P}_k$ is reinstated here) via the propagation equation (2.1). This information together with Z_k and $\mathscr{P}_k$ is denoted $\mathscr{P}_{k+1}$, which is all the prior information available at time t_{k+1}. The prior pdf of $\underline{x}_{k+1}$ may be written:

$$p\left(\underline{x}_{k+1} \middle| \mathscr{P}_{k+1}\right) = \int p\left(\underline{x}_{k+1} \middle| \underline{x}_k\right) p\left(\underline{x}_k \middle| \mathscr{P}_{k+1}\right) \underline{dx}_k \quad . \qquad (2.20)$$

$p\left(\underline{x}_{k+1} \middle| \underline{x}_k\right)$ is defined by the state propagation equations, and the second term:

$$p\left(\underline{x}_k \middle| \mathscr{P}_{k+1}\right) = p\left(\underline{x}_k \middle| Z_k , \mathscr{P}_k\right)$$

since the extra information on state propagation from t_k to t_{k+1} does not contribute to the pdf of state at t_k. Substituting equation (2.19) into equation (2.20) and performing the integrations (see Appendix A.3) gives:

$$p\left(\underline{x}_{k+1} \middle| \mathscr{P}_{k+1}\right) = \sum_{\ell=1}^{n_k} \Pr\left\{\mathscr{H}_{k\ell} \middle| \mathscr{P}_{k+1}\right\} p\left(\underline{x}_{k+1} \middle| \mathscr{H}_{k\ell} , \mathscr{P}_{k+1}\right) \qquad (2.21)$$

where $\Pr\left\{\mathscr{H}_{k\ell} \middle| \mathscr{P}_{k+1}\right\} = \beta_{k\ell}$

and $p\left(\underline{x}_{k+1} \middle| \mathscr{H}_{k\ell} , \mathscr{P}_{k+1}\right) = \mathscr{N}\left(\underline{x}_{k+1} ; \bar{\underline{x}}_{k+1\ \ell} , M_{k+1\ \ell}\right)$,

with

$$\bar{\underline{x}}_{k+1\ \ell} = \Phi \hat{\underline{x}}_{k\ell}$$

and

$$M_{k+1\ \ell} = \Phi P_{k\ell}\Phi^T + \Gamma Q \Gamma^T .$$

The pdf (2.21) is of the same form as equation (2.4): it is a fully specified Gaussian mixture. Hence the initial supposition of section 2.3.1 is proved by induction.

2.4 Discussion

It has been shown that the posterior pdf of the target state, just after incorporating the latest set of measurements, is a Gaussian mixture given by equation (2.19). The recursive procedure required to obtain this result is shown in the flow diagram, Fig 2.1. This procedure constitutes the optimal tracking filter for the problem stated in section 2.2. The Gaussian mixture (2.19) is a complete description of the filter's knowledge of the target state at time step k . Each component of the mixture represents a potential target track and is a Kalman filter estimate of the state vector based on a possible history of true and false measurements. At time t_k the n_k components represent *all feasible track histories*. The weighting $\beta_{k\ell}$ is the probability that track history ℓ is the correct one.

The pdf of target state contains all the available information so that, in principle, *an optimal estimate* based on any desired criterion may be obtained. For example the minimum mean square error estimate is the mean of the distribution (see Jazwinski[27]). From equation (2.19) the posterior mean of $\underline{x}_k$ is given by:

$$\hat{\underline{x}}_k = \sum_{\ell=1}^{n_k} \beta_{k\ell}\, \hat{\underline{x}}_{k\ell} , \qquad (2.22)$$

which is a weighted sum of the mean state vectors corresponding to each possible track history. Also the covariance of this estimate may be obtained from (2.19) (see Appendix B):

$$P_k = \sum_{\ell=1}^{n_k} \beta_{k\ell}\left(P_{k\ell} + \underline{\hat{x}}_{k\ell}\, \underline{\hat{x}}_{k\ell}^T\right) - \underline{\hat{x}}_k\, \underline{\hat{x}}_k^T \quad . \tag{2.23}$$

The mean may not be the most useful estimate for the state vector and in any case, a single value of $\underline{x}_k$ is a somewhat inadequate summary of a mixture distribution, especially if there are significant well spaced components.

For most interesting cases, the number of components n_k rapidly becomes very large with increasing k (see equation (2.19)). This rapid growth in the number of components may be viewed as a branching process. For instance, suppose that at time step $k-1$, the mixture distribution comprises two components. So there are two feasible tracks which are projected forwards to time step k. Suppose that at this time two measurements $\underline{z}_{k1}$ and $\underline{z}_{k2}$ are received. There are three possibilities:

$$\Psi_{k0} : \underline{z}_{k1} \text{ and } \underline{z}_{k2} \text{ are false ,}$$

$$\Psi_{k1} : \underline{z}_{k1} \text{ is true and } \underline{z}_{k2} \text{ is false ,}$$

or

$$\Psi_{k2} : \underline{z}_{k1} \text{ is false and } \underline{z}_{k2} \text{ is true .}$$

Thus the two feasible tracks from the previous time step may each be updated three different ways, giving rise to six feasible tracks at time step k (see Fig 2.2). Since every component must be propagated at

each time step, implementation of the optimal filter is impractical, and to proceed approximations must be imposed. This is the subject of the next chapter.

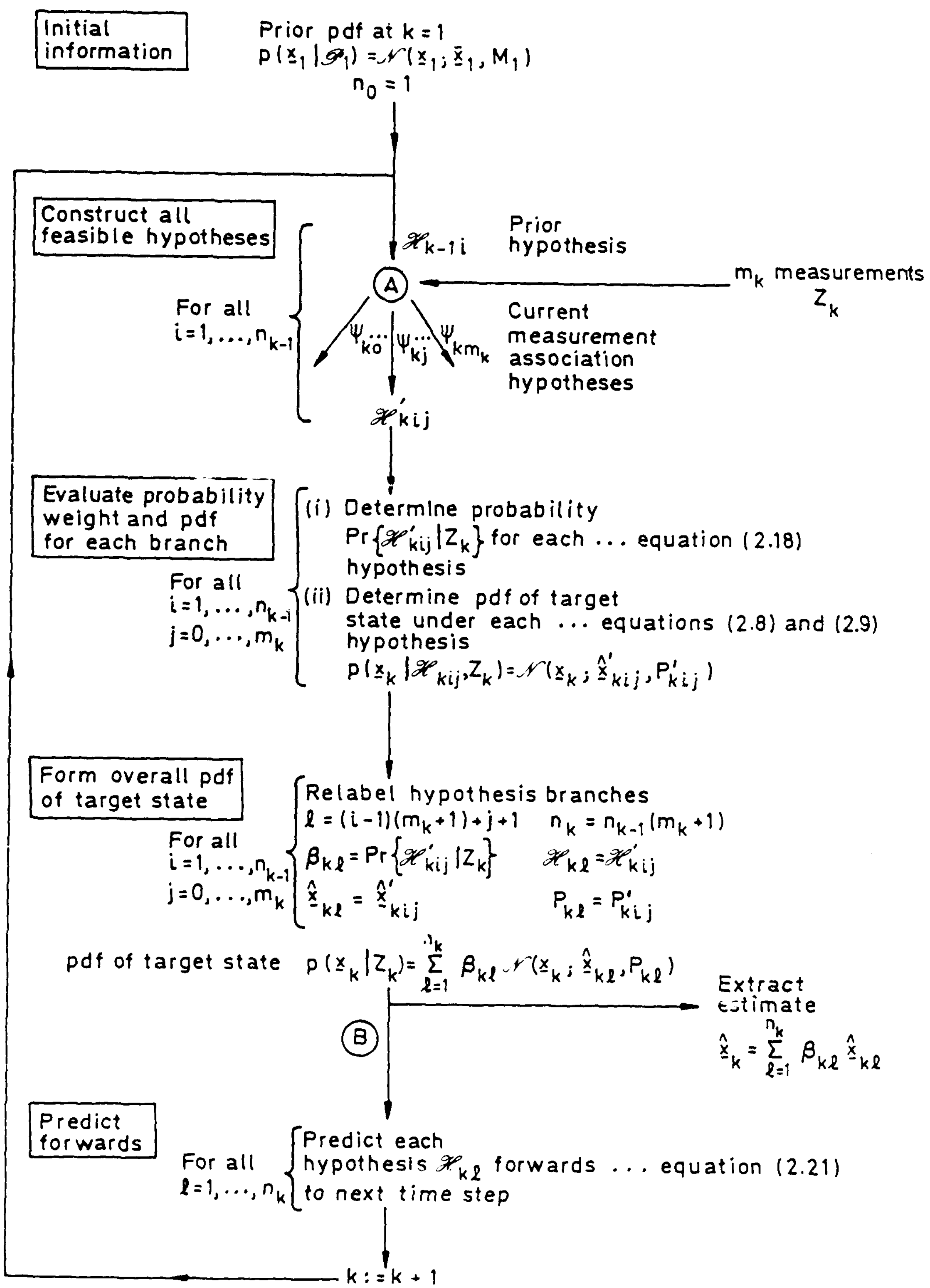


Fig 2.1 The optimal tracking filter for the baseline problem

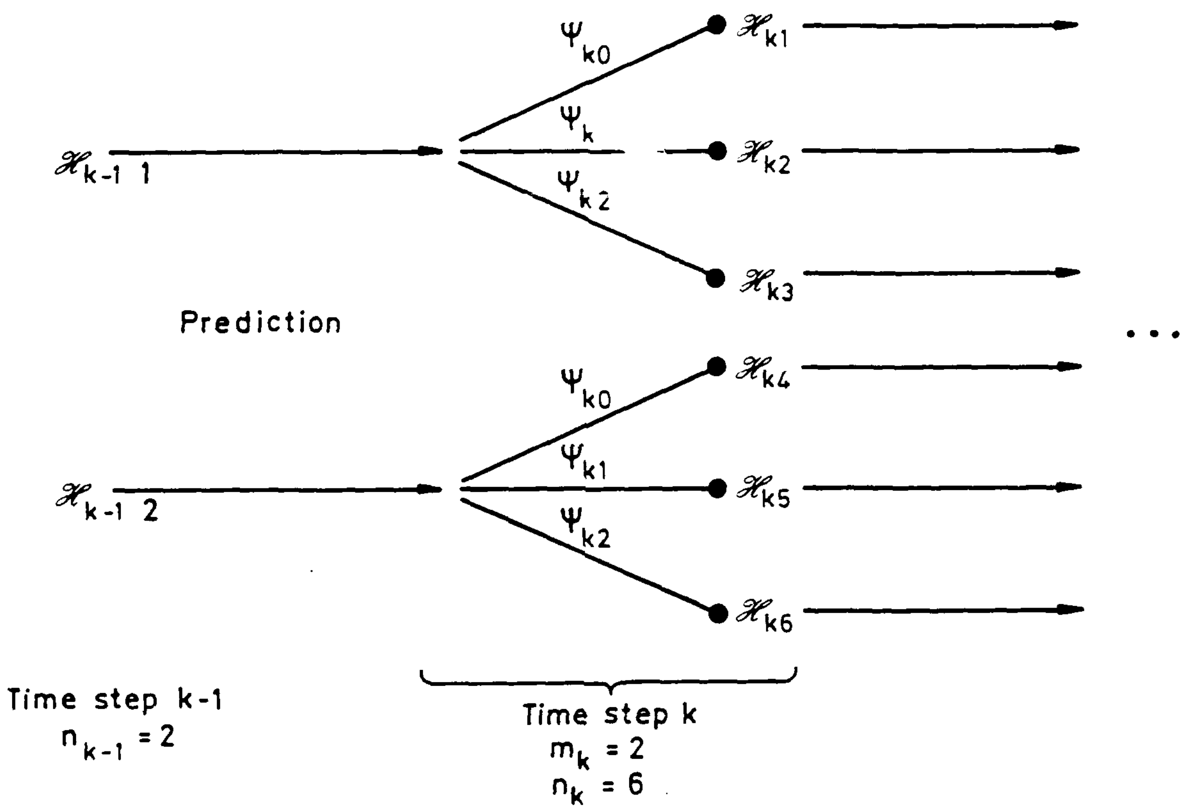


Fig 2.2 Growth of hypotheses or feasible tracks

3 CONTROLLING THE GROWTH OF MIXTURE COMPONENTS

3.1 Introduction

To implement the tracking filter described in the previous chapter, it is essential to control the growth of the number of components in the Gaussian mixture (equation (2.19)) at every time step. The maximum number of components that can be allowed, depends on the computing power (in terms of storage and speed of operation) and the time available to perform the calculations of the filter recursions. The maximum number N_T of components allowed in the mixture after approximation should be chosen so that the probable increase in the number of components from measurements received at the following time step is within the capability of the processor. If the growth exceeds this capability, the posterior pdf (2.19) may be truncated in an arbitrary fashion, rather than be subject to a considered approximation.

In this study, control of the growth of hypotheses is achieved in two stages at each time step. Firstly a coarse acceptance test is applied (see section 3.2), which rejects any hypothesis that appears to be very unlikely, on the basis of prior information. This control is applied at point A on the filter flow diagram, Fig 2.1. This test is computationally inexpensive as the unlikely hypotheses are rejected before their corresponding posterior mixture components need be evaluated. Hopefully the effect of this acceptance test on the posterior distribution will be insignificant. Since it is quite likely that the number of components left will still be excessive, further reduction may be necessary. This is applied at point B on Fig 2.1, which is after the posterior mixture distribution of the target state has been compiled from all hypotheses that have passed the coarse

acceptance test. The second stage is to approximate the mixture distribution and so reduce the number of its components from a posterior point of view (*ie* after filter update). To reduce the number of components below the specified limit, N_T, it may be necessary to make significant modifications to the distribution, and so careful consideration should be given to the design of this approximation method as it will affect filter performance.

What we require from a mixture reduction algorithm is discussed in section 3.3 and reported methods for such approximations are reviewed in section 3.4. It is argued that these reported techniques do not adequately fulfil our requirements and so two new approximation algorithms have been developed (see sections 3.6 and 3.7). The performance of these reduction algorithms for a tracking problem is assessed by simulation in the following chapter.

3.2 Coarse acceptance test

Each component of the posterior pdf (2.19) of target state is generated by updating a feasible track from the prior pdf with either one of the received measurements or by prediction on the assumption that all received measurements are false (see section 2.3). It is most convenient to generate equation (2.19) by considering each feasible prior track in turn, and evaluating all the possible posterior tracks which spring from that branch. Consider the prior track, or component i of equation (2.4), that corresponds to hypothesis $\mathscr{H}_{k-1\,i}$. The prior pdf of the true measurement under $\mathscr{H}_{k-1\,i}$ is the Gaussian with mean $H\bar{\underline{x}}_{ki}$ and covariance S_{ki}. From knowledge of this distribution, an acceptance or validation region in the measurement space can be defined, such that under hypothesis $\mathscr{H}_{k-1\,i}$, the probability of the true measurement (if it is detected) falling outside the region is very small. (This type of acceptance

test is commonly applied to measurement-track association problems where ambiguities may exist - see Blackman[1]). If the validation region is chosen so that the probability density of the true measurement at any point within the region exceeds that at all points outside the region, then since the distribution is Gaussian, the acceptance region is the interior of a hyperellipsoid. Thus a measurement $\underline{z}_{kj}$ is accepted for updating hypothesis $\mathscr{H}_{k-1\ i}$ if and only if:

$$\left(\underline{z}_{kj} - H\bar{\underline{x}}_{ki}\right)^T S_{ki}^{-1}\left(\underline{z}_{kj} - H\bar{\underline{x}}_{ki}\right) < T_A \quad . \tag{3.1}$$

Note that since the false measurements have a uniform distribution, this is equivalent to subjecting each measurement to a likelihood ratio test. For a true measurement $\underline{z}_{kj}$, under hypothesis $\mathscr{H}_{k-1\ i}$, the LHS of (3.1) is a sample from a χ^2 *distribution with* number of degrees of freedom equal to the dimension of $\underline{z}_{kj}$. So once the acceptable probability P_M of missing the true measurement (if the target is detected) under hypothesis $\mathscr{H}_{k-1\ i}$ has been chosen, the required value of the threshold T_A may be obtained from tables of χ^2 .

The acceptance test has been used in all the simulations of this study. To avoid any significant performance degradation as a result of the acceptance test, P_M was set to the very small value of 0.001. This corresponds to $T_A = 13.82$ for two dimensional measurement space. Also to take account of the possibility of rejecting the true measurement, the detection probability P_D should be replaced by $P_D(1-P_M)$ in equation (2.18). Thus, when the acceptance test is employed, even if $P_D = 1$, a component is generated for the finite probability of missing the true measurement.

3.3 Requirements of a mixture reduction algorithm

The following criteria have been identified for the design of a mixture reduction algorithm:

(i) The approximation should result in another Gaussian mixture. This is necessary to allow the tracking filter algorithm to be implemented as a bank of Kalman filters.

(ii) The algorithm should allow the maximum number N_T of components after approximation to be specified.

(iii) Whenever possible, reduction should be achieved without modifying the 'structure' of the distribution beyond some acceptable limit. Conversely, to avoid retaining unnecessary components, reduction should continue until this limit is reached, so that the approximation may contain less than N_T components. Note that this criterion is in terms of mixture structure modification because it is feasible to define and compute such a measure. Also it is likely that the extent of modification is related to practical performance measures, such as the probability of losing track, which cannot be readily computed as a function of mixture approximation.

(iv) Intuitively, the approximation should preserve the mean and covariance of the original mixture. Unfortunately, after propagation of the approximated mixture via the filter update and prediction relations, the mean and covariance of the updated mixture will not, in general, coincide with those of the optimal solution.

(v) The reduction algorithm should be computationally efficient (reduction must be accomplished within the filter update period), even when the original mixture consists of a large number of components (for example over 100), each with a different covariance matrix.

3.4 Review of mixture reduction techniques

A number of techniques for controlling the growth of the mixture distribution have been reported. The simplest method is to reduce the mixture to a single Gaussian component at each time step, and the crudest means of achieving this is to choose that mixture component corresponding to the most probable hypothesis. When the probability of detection is unity, this corresponds to the nearest neighbour approach, *ie* update the track by using the measurement $\underline{z}$ which minimizes the expression:

$$\left(\underline{z} - H\bar{\underline{x}}_k\right)^T S_k^{-1}\left(\underline{z} - H\bar{\underline{x}}_k\right) \quad .$$

However this technique takes no account of the possibility that the wrong hypothesis may have been chosen and results in what is essentially a decision-directed filter (see Bar-Shalom[3]). A considerable improvement on this method is the Probabilistic Data Association Filter (PDAF)[11], or probabilistic editor[37], in which the single Gaussian approximation is chosen to match the mean and covariance of the full posterior mixture (see Appendix B, equations (B-5) and (B-6)). Thus the hypotheses are effectively combined and the uncertainty is recognized in the covariance of the approximating Gaussian. The PDAF has been promoted principally by Bar-Shalom and it may be thought of as a lower bound on the range of possible approximations meeting requirement (iv) (the mean and covariance are preserved); the upper bound being

obtained when all components are retained. The PDAF does not meet requirements (ii) or (iii). The filter performs well in a number of cases (see Ref 15) and is computationally very economical. However in many circumstances, the single Gaussian approximation will destroy important structure in the mixture distribution, especially when a number of well spaced components are present. In this case it should be better to consider approximations which retain several components.

Singer *et al*[10] have developed an N-scan filter in which components of the mixture distribution are combined according to the history of hypotheses. If several components result from updating by the same measurements over the last N-scans, then the components are combined. For a particular N, the performance of this method is likely to depend on the responsiveness of the filter to incoming measurements. This in turn depends on the covariances Q and R. If the filter is very responsive, components with the same measurement history over recent scans will be very similar and the consequent performance penalty in combining these components should be small. A disadvantage is that the number of components retained is not limited (requirement (ii)). However provided N is small, the algorithm should be computationally efficient: no measures of similarity need be calculated, although the recent history of measurement acceptance must be stored. For the simulation example reported in Ref 10, near optimal performance is claimed for only a single scan memory.

Gaussian mixture distributions with an increasing number of components also occur in system switching problems, where the parameters of the system are subject to abrupt changes or jumps. Thus approximation techniques have also been developed to implement filters for these problems (see the survey by Pattipati and Sandell[38]). A technique

known as the generalized pseudo Bayes algorithm (GPBA) has been developed by Jaffer and Gutpa[39], which is the equivalent of the N-scan memory filter. In a similar vein, Blom[40] has developed an interacting multiple model (IMM) algorithm in which components of the prior distribution are merged before measurement update. The special case of the GPBA where only a single Gaussian is propagated is called the pseudo Bayes method, and this is the equivalent of the PDAF. The pseudo Bayes method was proposed by Ackerson and Fu[41], although they omitted the 'between components' contribution to the covariance of the approximating Gaussian (see next section).

The remainder of the methods described in this section are direct approximations of the posterior mixture distribution, without reference to the measurement history, which allow more than one component to be retained. All of these techniques involve merging or discarding components of the mixture. The simplest of these schemes is to retain only the N most probable components at each time step (see Tugnait[42]). A refinement of this method suggested in Ref 43 is to combine components which are close in the sense of the Bhattacharyya distance measure (see below) before rejecting components; but this does not appear to have been implemented.

Alspach[25] and Lainiotis and Park[44] have suggested schemes in which the mixture is approximated by merging and pruning operations, none of which exceed a specified penalty measure. Alspach defines the penalty of approximating the mixture $p(\underline{x})$ by $p_A(\underline{x})$ as the Kolmogorov variational distance between the two distributions:

$$\kappa = \int | p(\underline{x}) - p_A(\underline{x}) | \, d\underline{x} \quad .$$

Weiss, Upadhyay and Tenney[45] also analyse the penalty of merging components in terms of κ. Lainiotis and Park use a penalty measure based on the Bhattacharyya coefficient ρ, which is defined by:

$$\rho = \int \sqrt{p(\underline{x})\ p_A(\underline{x})}\ d\underline{x} \ .$$

ρ lies between zero and one, and $\rho = 1$ if $p(\underline{x}) = p_A(\underline{x})$. Thus $1 - \rho$ is a measure of the penalty of approximating $p(\underline{x})$ by $p_A(\underline{x})$. These distance measures are related by (see Kailath[46]):

$$\sqrt{1 - \rho^2} \geqslant \frac{\kappa}{2} \geqslant 1 - \rho \ .$$

Bounds on these penalty measures in terms of the mixture parameters have been derived for deleting a component and for merging a pair of components (see Refs 25 and 44). The authors suggest that fixed acceptable penalty levels should be chosen and that the mixture should be reduced by merging and pruning operations which do not exceed these penalty levels. The method of Lainiotis and Park would require the calculation of the Bhattacharyya coefficient between every pair of mixture components. This would be very time consuming and the method does not appear to have been implemented. The method of Alspach assumes that the covariance of all components is the same. This situation is maintained as filtering proceeds by ignoring the between component contribution to the covariance of the merged components, and so overall covariance is not preserved with this method.

The mixture reduction techniques derived in the following sections may be viewed as developments of these direct approximation methods. The new algorithms, which are essentially merging operations, cater for components with different covariances, and the maximum number

of components after approximation may be chosen as required. Also, at each time step, the overall mean and covariance are preserved, save for certain insignificant components which may be discarded. The algorithms are based on the premise that changes to the 'structure' of the mixture should be minimized. The measure of structure is derived from a decomposition of the mixture covariance matrix.

3.5 Mixture structure: the covariance matrix

Consider any N-component mixture distribution with pdf:

$$p(\underline{x}) = \sum_{i=1}^{N} \beta_i p_i(\underline{x})$$

where $p_i(\underline{x})$ is a component pdf

and β_i is a probability associated with the ith component such that:

$$\beta_i > 0$$

and

$$\sum_{i=1}^{N} \beta_i = 1 \quad .$$

The covariance matrix P of this mixture may be decomposed into two contributions, W and B (see Appendix B, equation (B-3)):

$$P = W + B$$

where $W = \sum_{i=1}^{N} \beta_i P_i$

$$B = \sum_{i=1}^{N} \beta_i \left(\hat{\underline{x}}_i - \hat{\underline{x}}\right)\left(\hat{\underline{x}}_i - \hat{\underline{x}}\right)^T$$

$$\hat{\underline{x}} = \sum_{i=1}^{N} \beta_i \hat{\underline{x}}_i$$

is the mean of the distribution and $\hat{\underline{x}}_i$ and P_i are the mean and covariance of the ith component. The matrix W may be interpreted as the contribution from the covariance 'within' each component of the mixture and it depends on the spread of each individual component. B may be interpreted as the between component contribution which is due to the separation between the mixture components. B and W are both symmetric matrices, W being positive definite and B being positive semidefinite.

Suppose that the mixture distribution is approximated by merging several components together. If $\mathbb{C}$ is the set of subscripts of components to be merged, then the probability mass of the new component is:

$$\beta' = \sum_{i \in \mathbb{C}} \beta_i \quad . \tag{3.2}$$

To preserve the overall mean of the mixture (requirement (iv)):

$$\hat{\underline{x}} = \sum_{i=1}^{N} \beta_i \hat{\underline{x}}_i = \beta' \hat{\underline{x}}' + \sum_{i \notin \mathbb{C}} \beta_i \hat{\underline{x}}_i \quad ,$$

so that the mean of the new component is given by:

$$\underline{\hat{x}}' = \frac{1}{\beta'} \sum_{i \in \mathbb{C}} \beta_i \, \underline{\hat{x}}_i \quad . \tag{3.3}$$

Also to preserve the overall covariance, from equation (2.23):

$$P = \sum_{i=1}^{N} \beta_i \left(P_i + \underline{\hat{x}}_i \, \underline{\hat{x}}_i^T \right) - \underline{\hat{x}} \, \underline{\hat{x}}^T$$

$$= \sum_{i \notin \mathbb{C}} \beta_i \left(P_i + \underline{\hat{x}}_i \, \underline{\hat{x}}_i^T \right) + \beta' \left(P' + \underline{\hat{x}}' \, \underline{\hat{x}}'^T \right) - \underline{\hat{x}} \, \underline{\hat{x}}^T \quad ,$$

so that the covariance of the new component is given by:

$$P' = \frac{1}{\beta'} \sum_{i \in \mathbb{C}} \beta_i \left(P_i + \underline{\hat{x}}_i \, \underline{\hat{x}}_i^T \right) - \underline{\hat{x}}' \, \underline{\hat{x}}'^T \quad . \tag{3.4}$$

Although the overall covariance P is unchanged, this merging of components results in a loss of between components covariance B which is balanced by an increase in W. To see this, let W' and B' be the within and between covariances of the approximated mixture. Since overall covariance is preserved:

$$P = W + B = W' + B' \quad . \tag{3.5}$$

Thus the matrix L defined as:

$$L = B - B' \quad ,$$

is given by:

$$L = W' - W$$

$$= \beta' P' - \sum_{i \in \mathbb{C}} \beta_i P$$

$$= \sum_{i \in \mathbb{C}} \beta_i \, \hat{\underline{x}}_i \, \hat{\underline{x}}_i^T - \beta' \, \hat{\underline{x}}' \, \hat{\underline{x}}'^T$$

$$= \sum_{i \in \mathbb{C}} \beta_i \left(\hat{\underline{x}}_i - \hat{\underline{x}}'\right)\left(\hat{\underline{x}}_i - \hat{\underline{x}}'\right)^T \quad , \tag{3.6}$$

which is a positive semi-definite matrix. This shift of covariance from B to W is a rough measure of the change in the structure of a mixture distribution when components are combined. (Techniques have been developed for Cluster Analysis using a similar decomposition of the data scatter matrix (see Hand[47]).)

3.6 The Joining Algorithm

3.6.1 Derivation

Ideally the final partition of components into sets for merging, should be such that the increase in some cost function is minimized. However to reduce the mixture from N to M components, this could involve the evaluation of the criterion for every possible partition to identify the minimum. Such a procedure for a number of different values of M would be far too time consuming and so a suboptimal approach has been adapted from the agglomerative methods of Cluster Analysis (see Hand[47]). In this approach, which we call the Joining Algorithm, a pair of components are merged at every iteration of the algorithm. The components for merging are chosen to minimize the incease in the chosen criterion at each stage. Clearly there is no

guarantee that the final partition from such a procedure will achieve the smallest possible value of the cost function.

To implement the Joining Algorithm using a cost function based on an increase in the within component covariance, we require a suitable scalar measure. From equations (3.3) and (3.6), if components i and j are merged, the increase in W is given by:

$$L_{ij} = \frac{\beta_i \beta_j}{\beta_i + \beta_j}\left(\hat{\underline{x}}_i - \hat{\underline{x}}_j\right)\left(\hat{\underline{x}}_i - \hat{\underline{x}}_j\right)^T \quad . \tag{3.7}$$

One possible measure is the trace of L_{ij}, which is the squared Euclidean distance between the component means modified by the factor $\beta_i \beta_j/(\beta_i + \beta_j)$. However this has the disadvantage that it is dependent on the scaling of the elements of the state vector and so is problem dependent. This difficulty is avoided by using the Mahalanobis distance (see Ref 47) to give:

$$d^2_{ij} = \frac{\beta_i \beta_j}{\beta_i + \beta_j}\left(\hat{\underline{x}}_i - \hat{\underline{x}}_j\right)^T P^{-1}\left(\hat{\underline{x}}_i - \hat{\underline{x}}_j\right) , \tag{3.8}$$

where P is the covariance of the whole mixture. This distance measure is related to L_{ij} by:

$$d^2_{ij} = \mathrm{tr}\left(P^{-1} L_{ij}\right) \quad .$$

This measure is invariant under all non-singular linear transformations of the state vector. At each iteration of the Joining Algorithm, the two components which are closest in the sense of the distance measure equation (3.8) are combined to form a new component defined by equations (3.2) to (3.4).

The minimum value of the distance measure at each iteration is an indicator of the change in distribution structure resulting from the merging of the two closest components. It is shown in Appendix C that this minimum distance increases monotonically as reduction proceeds, and so each merging operation increases this measure of structural modification. (Distance measures with this property are said to be not subject to reversals - see Anderberg[48], page 141.) Thus if a threshold defining the acceptable modification to the distribution is specified, approximation should proceed until the minimum distance exceeds this threshold. For convenience we compare the squared distance d_{ij}^2 with a threshold T. In choosing a value for the threshold T, it is useful to note that the squared distance d_{ij}^2 is bounded. To see this (from equations (3.5) and (3.6)):

$$\begin{aligned} P &= W + B = W + B' + B - B' \\ &= (W + B') + L_{ij} \,, \end{aligned}$$

where P and W are positive definite $n \times n$ matricies, and B' and L_{ij} are positive semi-definite. Multiply through by P^{-1} to give:

$$I = P^{-1}(W + B') + P^{-1} L_{ij} \,.$$

Taking the trace gives:

$$n = \mathrm{tr}\left[P^{-1}(W + B')\right] + \mathrm{tr}\left[P^{-1} L_{ij}\right] \,.$$

Hence since P^{-1} and $(W + B')$ are both positive definite,

$$d_{ij}^2 < n \,.$$

Note that for our tracking problem, n is the dimension of the state space. Thus we have chosen T to be a constant fraction of this upper bound n. Simulation studies indicate that a value of:

$$T = 0.001\,n$$

retains sufficient components to give, on visual inspection, a good approximation to the mixture.

At each iteration, the algorithm determines the number N_R of remaining components, excluding the set of smallest components with total probability mass (*ie* the sum of their β weights) less than B_T. If d^2_{ij} exceeds T before N_R has been reduced below the specified maximum N_T, then approximation continues beyond the acceptable limit of modification. The purpose of B_T, which has been set to 0.01, is to avoid wasting effort on grouping insignificant components. A flow diagram of the Joining Algorithm is given in Fig 3.1.

3.6.2 An example of mixture reduction with the Joining Algorithm

The Joining Algorithm has been applied to a four-dimensional Gaussian mixture distribution taken from the tracking simulation of Chapter 4. For illustration, the distribution is only shown as a function of two dimensions x and y, which are the Cartesian co-ordinates of the target position. Fig 3.2 gives a perspective view of the pdf of the original mixture, while in Fig 3.3 it is shown as a contour plot with logarithmic contour spacing to bring out the shape of the smaller components. The distribution is composed of 37 components.

The final partition of components produced by the Joining Algorithm with $N_T = 10$ is shown in Fig 3.4. In this figure the means of the original components are plotted as numbers which denote the final component to which the original is assigned. The final components are ordered according to decreasing probability mass, so component number 1 has the largest β weight. The original components are colour coded according to their β weights as indicated on the diagrams. The actual position of the target is also shown; it is close to the means of two of the larger original components. Note that after reduction the maximum permitted number of components, *ie* ten, has been retained, indicating that the minimum squared distance measure has exceeded the acceptable modification threshold T . The grouping of components shown in Fig 3.4 appears to be consistent with maintaining, as far as possible, the structure of the distribution, although it should be noted that the distribution is four-dimensional and only two of these dimensions are shown here. The mixture approximation corresponding to this partition is shown in Fig 3.5; it appears to be an excellent approximation of the original (Fig 3.2).

If N_T is reduced to four, the components are further merged to produce the partition shown in Fig 3 6. Here the original central concentration of components has been split into three groups, of which number 3 includes one of the less significant remote concentrations. The mixture approximation for $N_T = 4$ is shown in Fig 3.7. Comparing this with Fig 3.2, it can be seen that the original mixture has been significantly modified.

The history of how components are merged together for this example is illustrated by the tree diagram of Fig 3.8. The mean (x and y elements only) and the β weight of each of the original mixture

components are listed on the left hand side of this diagram. The tree structure which grows from these components indicates which components were merged together and at what joining distance* this occurred. Since the joining distance always increases (as shown in Appendix C), the sequence in which components were combined is the same as the ordering of the merging from left to right in the diagram. For this reason it is always possible to arrange the original components so that none of the branches of the tree cross one another. Note that the joining distance is plotted on a logarithmic scale and that all components with β weights less than 0.001 have been merged at least once before the joining distance has risen above $d^2 = 5 \times 10^{-4}$, which is only 0.0125% of the maximum possible joining distance $d^2 = 4$.

In this example the mixture could be reduced to 17 components without exceeding the joining distance threshold $T = 0.004$, but to achieve a reduction to 10 components, the final joining distance was $d^2 = 0.028$. The numbering of the branches at this stage on the diagram corresponds to the cluster numbers of Fig 3.4, so the clusters are numbered according to decreasing probability mass. To further reduce the mixture to only four components, the joining distance increased to about 0.3, which is 75T. The branch numbers at this stage corresponds to the cluster numbers of Fig 3.6. If merging continues until only one component remains, the single Gaussian PDAF approximation of the mixture is produced. The final merging is at a distance $d^2 = 0.703$, within the theoretical limit of 4. (Further examples of the final joining distance for $N_T = 10$ and $N_T = 4$ are given in the next section.)

---

* We loosely refer to joining distance although this is actually the squared measure d_{ij}^2.

3.6.3 The control of mixture components for a tracking example

The Joining Algorithm (in conjunction with the coarse acceptance test) has been used to control the growth of mixture components in a simulation of target tracking using the filter described in Chapter 2. The tracking problem is specified in Chapter 4.

In Fig 3.9 the number of mixture components before and after reduction by the Joining Algorithm is shown for each time step during the tracking operation. Between time steps, the number of components increases according to the number of measurements passed by the coarse acceptance test. Also shown is the final joining distance at each time step. For this example the threshold T was exceeded on 42% of the time steps to achieve an acceptable reduction specified by $N_T = 10$. Note that when the final joining distance is below T, the number of components in the reduced mixture is usually less than N_T. When N_T is reduced to 4 (Fig 3.10), the final joining distance is almost always greater than T, and on average is about ten times larger than the average final joining distance for $N_T = 10$.

3.7 The Clustering Algorithm

3.7.1 Derivation

The second algorithm is based on the proposition that the mixture components with the largest β weightings carry the most important information. Thus starting with the largest component, this algorithm gathers in all surrounding components that are in some sense close to the principal component. Subsequently the largest component of the remainder is selected and the process is repeated until all the components have been clustered. This is called the Clustering Algorithm.

The distance measure chosen to represent the closeness of component i to the cluster centre is defined by:

$$D_i^2 = \frac{\beta_i \beta_c}{\beta_i + \beta_c}\left(\hat{\underline{x}}_i - \hat{\underline{x}}_c\right)^T P_c^{-1}\left(\hat{\underline{x}}_i - \hat{\underline{x}}_c\right) , \qquad (3.9)$$

where β_c , $\hat{\underline{x}}_c$ and P_c are the probability mass, mean and covariance of the principal component, and β_i and $\hat{\underline{x}}_i$ are the probability mass and mean of the ith component. This is the same as the distance measure d_{ij}^2 of the Joining Algorithm, except that the distance is normalized to the covariance of the cluster centre rather than the complete mixture. Indeed equation (3.7) is the motivation for the definition of D_i^2 . Note that D_i^2 is independent of the covariances of components being tested for clustering and that the selection of components for each cluster only involves the inversion of one symmetric matrix P_c . Any component i for which $D_i^2 < T_1$ is selected as a cluster member. The threshold T_1 defines the acceptable modification to the distribution.

In choosing T_1 , it is helpful to first consider the measure $D_i'^2$ defined by:

$$D_i'^2 = \left(\hat{\underline{x}}_i - \hat{\underline{x}}_c\right)^T P_c^{-1}\left(\hat{\underline{x}}_i - \hat{\underline{x}}_c\right) .$$

If the criterion for clustering a component i were $D_i'^2 < T_1'$, then any component i whose mean were to fall within the hyperellipsoid defined by T_1' would be clustered. This hyperellipsoid is a contour of constant probability density of the prinicipal component and the proportion of probability mass enclosed is a measure of the selectivity of the clustering operation. If T_1' were chosen so that only a small proportion, say 1%, of the probability mass of the cluster centre were

enclosed, then the structure of the distribution should be little altered by clustering. However $D_i'^2$ is independent of the probability mass β_i of the component, and intuitively, merging a large component would have a greater effect on the mixture than merging a small component. The modifying factor $\beta_i \beta_c/(\beta_i + \beta_c)$ biases this distance so that small components are more easily clustered while large components retain their individuality. It is suggested that the threshold for:

$$D_i^2 = \frac{\beta_i \beta_c}{\beta_i + \beta_c} D_i'^2 ,$$

should be chosen so that small components with β weights less than 0.05 are more readily clustered, while components with β weights exceeding 0.05 are clustered less readily. Fig 3.11 shows that the contour $\beta_i \beta_c/(\beta_i + \beta_c) = 0.05$ is close to the line $\beta_i = 0.05$ inside the region of interest, except when β_i is nearly equal to β_c. Thus it is suggested that to give a good mixture approximation, the threshold for D_i^2 should be set to:

$$T_1 = 0.05T_1' ,$$

where T_1' defines the hyperellipsoid containing only 1% of the probability mass. (T_1' can be found from tables of χ^2 with the number of degrees of freedom equal to the dimension of the statespace.)

Each cluster of components (some clusters may consist of a single component) is approximated by a single Gaussian defined by equations (3.2) to (3.4). Clustering proceeds until the probability mass of the unclustered components is less than B_T. As for the Joining Algorithm, the purpose of B_T, which is set to 0.01, is to avoid wasting effort on clustering insignificant components. If the number of clusters is

less than or equal to N_T, the unclustered components are deleted and approximation is complete; otherwise further reduction is necessary. This is achieved by repeating the clustering procedure on the first approximation including the unclustered components, but with the clustering threshold incremented by ΔT, *ie* $T = T_1 + \Delta T$. This clustering operation is iterated until the necessary reduction has been effected. The choice of the increment ΔT is a compromise between the number of iterations required and the possibility of clustering more components than necessary. In this study, the value of ΔT is fixed:

$$\Delta T = 0.05 \, \Delta T' \quad ,$$

where $T' + \Delta T'$ defines the hyperellipsoid which contains 6% of the probability mass of the principal component. Simulation work has shown this to be a reasonable compromise. For the simulation examples of this study, the statespace is four-dimensional, so from tables of χ^2 the algorithm thresholds have been set to:

$$T_1 = 0.01485$$

and

$$\Delta T = 0.02065 \quad .$$

Although ΔT is normally fixed, an override is provided which may increase the clustering threshold further to ensure that at least one component is clustered on each iteration. This mechanism is shown in the flow diagram of the algorithm given in Fig 3.12.

3.7.2 An example of mixture reduction with the Clustering Algorithm

The Clustering Algorithm has been applied to the same four-dimensional Gaussian mixture distribution that was used to demonstrate the operation of the Joining Algorithm (see Figs 3.2 and 3.3).

For $N_T = 10$, the final partition is shown in Fig 3.13 and the corresponding mixture approximation is shown in Fig 3.14. The approximation consists of nine components, although several algorithm iterations were required; *ie* the acceptable modification limit was exceeded. The composition of the final clusters is similar to the grouping produced by the Joining Algorithm (see Fig 3.4), although there are detailed differences. Also the mixture approximation is very similar to that produced by the Joining Algorithm (see Fig 3.5), and appears to be an excellent approximation of the original (see Fig 3.2).

The partition of components and the mixture approximation produced by the Clustering Algorithm with $N_T = 4$ are shown in Figs 3.15 and 3.16. The partition of the components is very similar to that of the Joining Algorithm with $N_T = 4$ (see Fig 3.6), the difference being the assignment of three components with β weights below 0.001 and one component with $0.01 < \beta < 0.1$. It is chiefly this one component which accounts for the obvious difference between the Clustering Algorithm approximation and the Joining Algorithm approximation (Fig 3.7) - also see the contour plot Fig 3.17. These approximations are significantly different from the original (Fig 3.2).

3.7.3 The control of mixture components for a tracking example

The Clustering Algorithm has been applied to mixtures generated by the same tracking example as used to exercise the Joining Algorithm in section 3.6.3. Fig 3.18 shows the number of components before and after reduction, the maximum clustering distance* threshold, and the number of algorithm iterations for each time step with $N_T = 10$.

* We loosely refer to clustering distance although this is actually the squared measure D_i^2 .

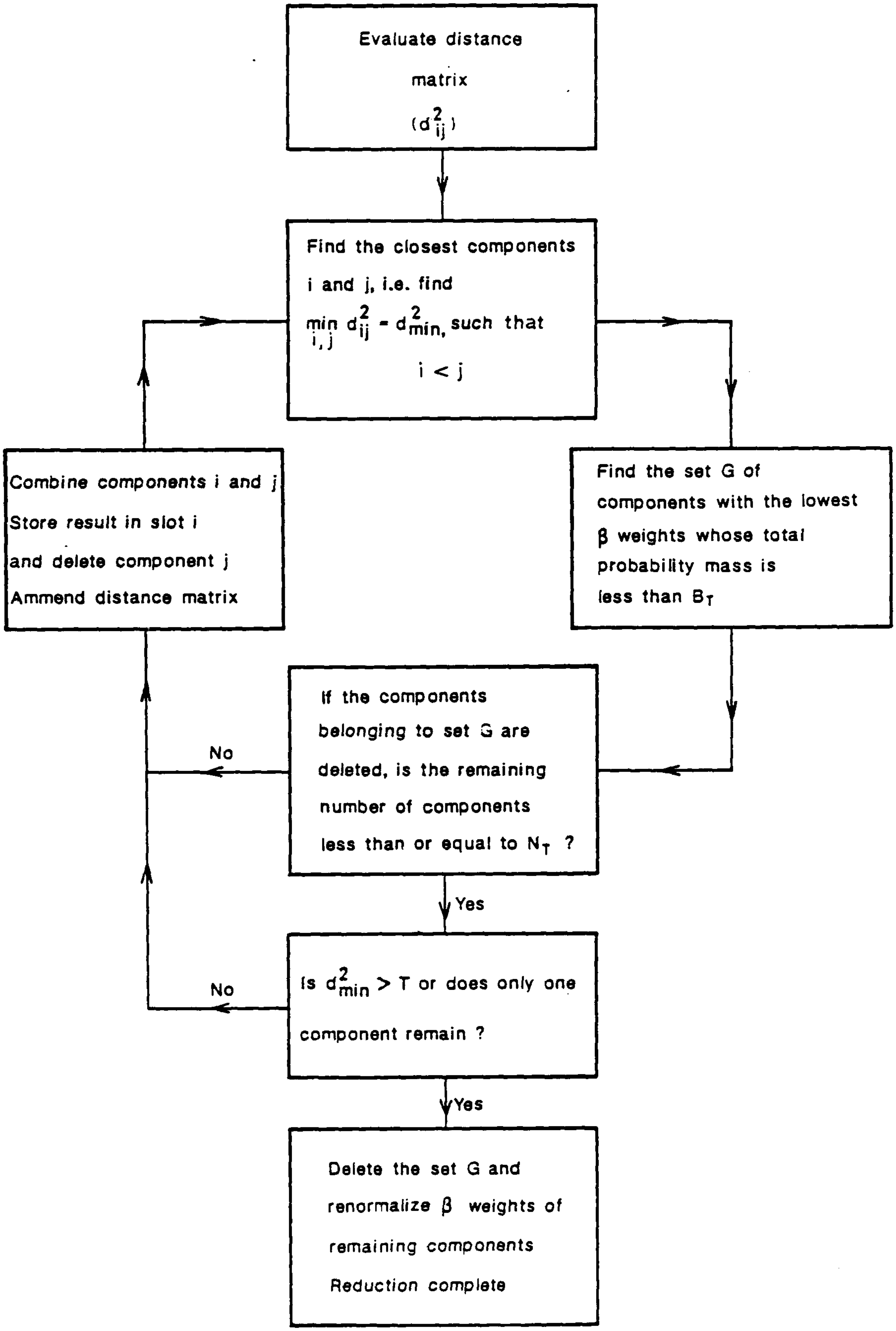
Adequate reduction to within ten components is achieved with a single algorithm iteration (that is with the threshold T_1) on 72% of the time steps, and no more than five iterations are ever required. Also comparing the plot of threshold value with the plot of the number of iterations, it can be seen that the override mechanism for increasing the threshold value by a jump in excess of ΔT has only been invoked on three time steps. When adequate reduction cannot be achieved without increasing the threshold above T_1, the number of components in the reduced mixture is never less than $N_T - 3$, showing that the algorithm did not merge many more components than necessary. Finally note that the plot of the number of components before and after reduction is similar to the corresponding plot for the Joining Algorithm (see Fig 3.9). Also the plots of the maximum joining distance and the clustering threshold show similarities.

Fig 3.19 shows the management of mixture components for $N_T = 4$. The work load of the Clustering Algorithm is considerably increased for this smaller value of N_T. Adequate reduction with a single iteration is achieved on only 22% of occassions, and a maximum of 11 iterations were required for one time step. However the override facility for increasing the threshold level was frequently employed, and without this feature the maximum number of iterations would have been close to 100. There is some similarity between the plot of number of components before and after reduction and the corresponding plot for the Joining Algorithm shown in Fig 3.10. However the match is not so good as for $N_T = 10$, showing that for small N_T the number of components is more sensitive to the reduction algorithm employed. This is probably because significant components have to be merged to achieve the necessary reduction.

3.8 Conclusions

Two new mixture reduction algorithms have been developed to meet a set of requirements for Bayesian tracking filters. These algorithms have been derived from the principle that the increase in the within component covariance should be minimized when components are merged. When applied to a Gaussian mixture distribution from a tracking example, excellent approximations can be achieved provided the number of N_T components allowed in the approximation does not force significant distinct components to merge. For small values of N_T, the approximations produced by the two algorithms were clearly different and some features of the original distribution were obviously blurred.

In Appendix D the computational requirements of the two reduction algorithms are analysed. It is shown that if the number of components before reduction is large compared with that after reduction, the number of operations required by the Joining Algorithm lies between the lower and upper bounds of the operation count for the Clustering Algorithm. Also for the Joining Algorithm a large distance matrix must be stored, while for the Clustering Algorithm storage requirements over those necessary to hold the mixture components are negligible. In the following chapter we compare the performance and the computation time of the two algorithms and the PDAF for an example of the baseline problem.



Evaluate distance
matrix
(d_{ij}^2)
Find the closest components
i and j, i.e. find
$\min_{i,j} d_{ij}^2 = d_{min}^2$, such that
$i < j$
Combine components i and j
Store result in slot i
and delete component j
Ammend distance matrix
Find the set G of
components with the lowest
β weights whose total
probability mass is
less than B_T
If the components
belonging to set G are
deleted, is the remaining
number of components
less than or equal to N_T ?
No
Yes
Is $d_{min}^2 > T$ or does only one
component remain ?
No
Yes
Delete the set G and
renormalize β weights of
remaining components
Reduction complete

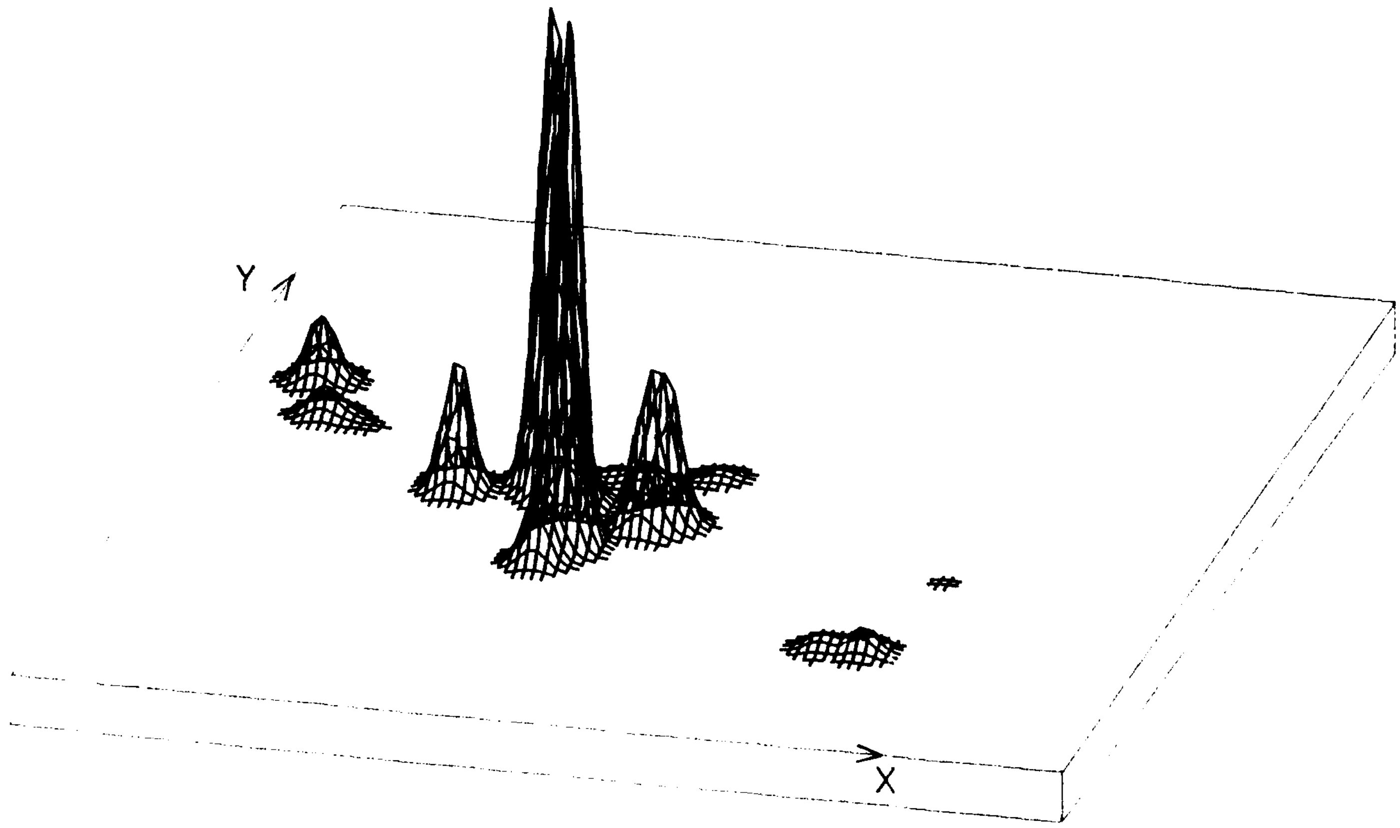


Fig 3.2 Mixture pdf before approximation (37 components)

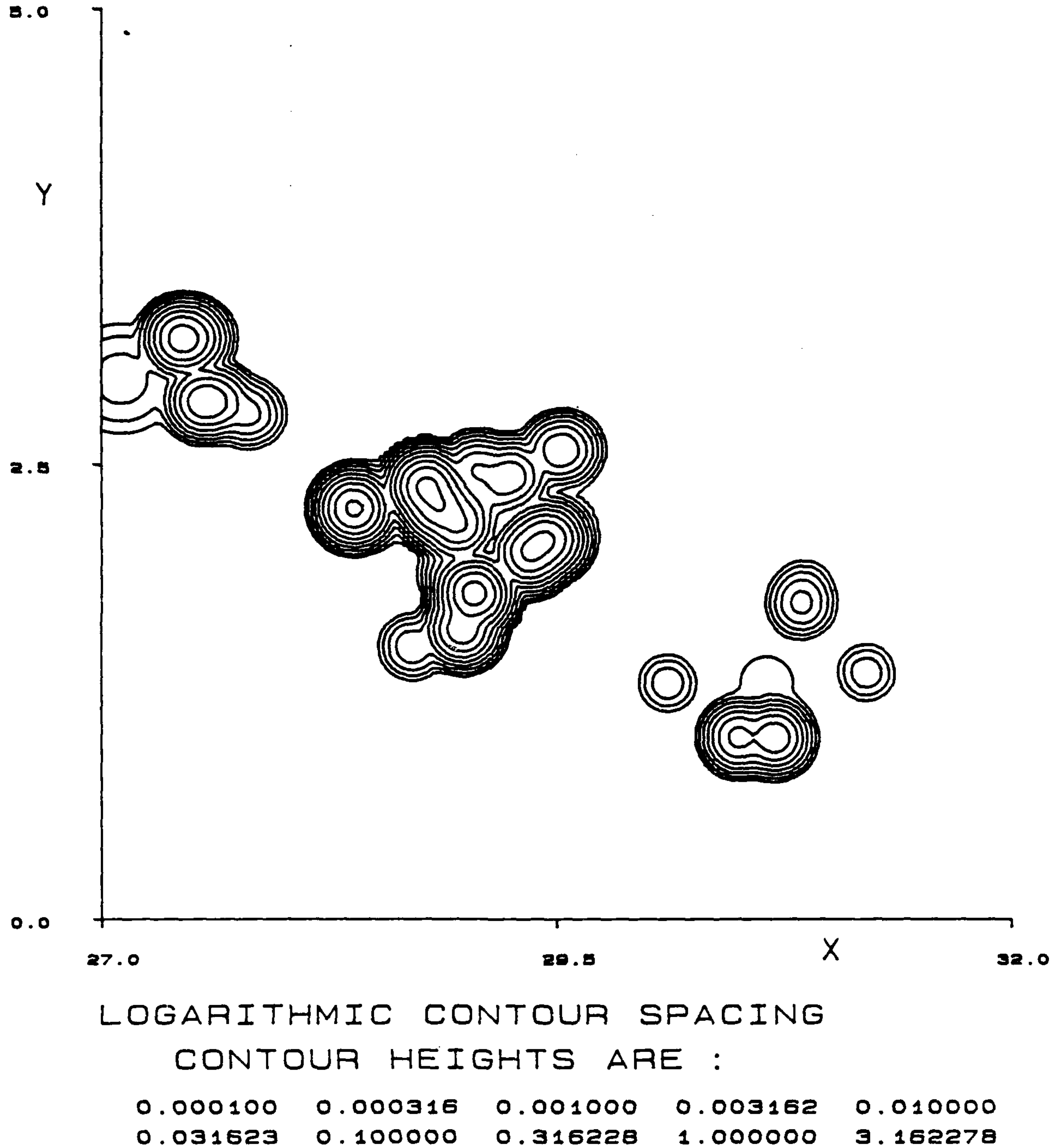


Fig 3.3 Mixture pdf before approximation (37 components)

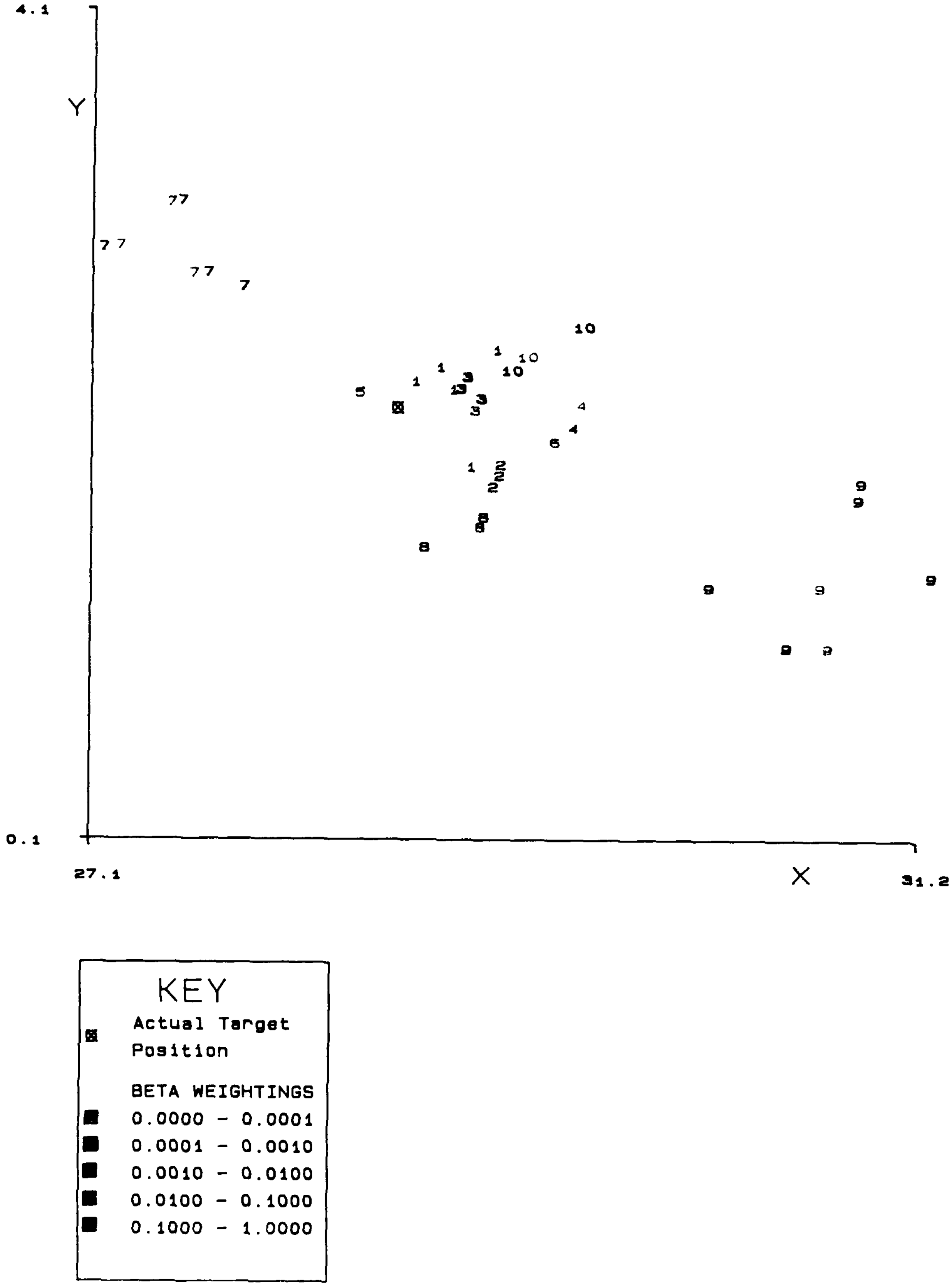


Fig 3.4 Partition of the original components by the Joining Algorithm for N_T = 10

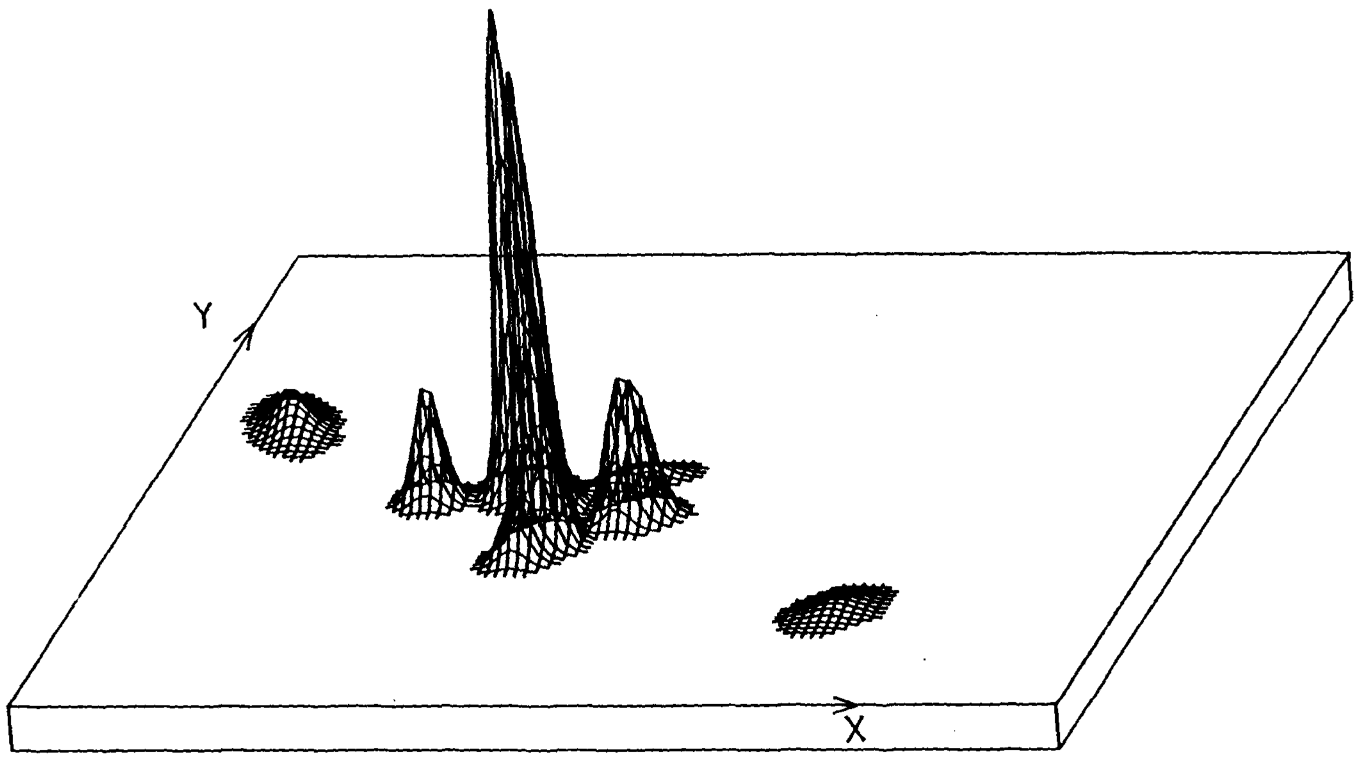


Fig 3.5 Approximated mixture pdf produced by the Joining Algorithm with $N_T = 10$ (10 components)

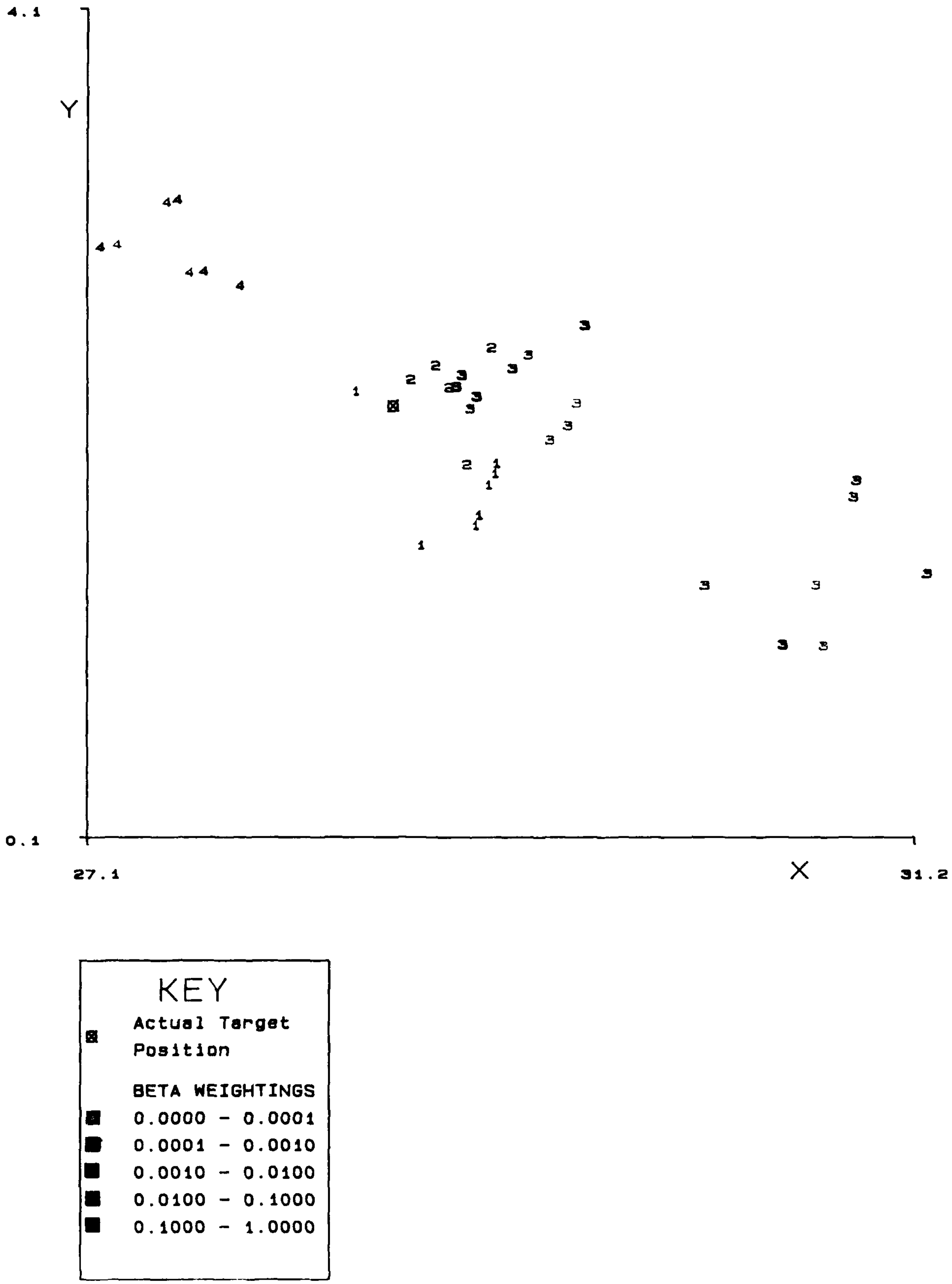


Fig 3.6 Partition of the original components by the Joining Algorithm for $N_T = 4$

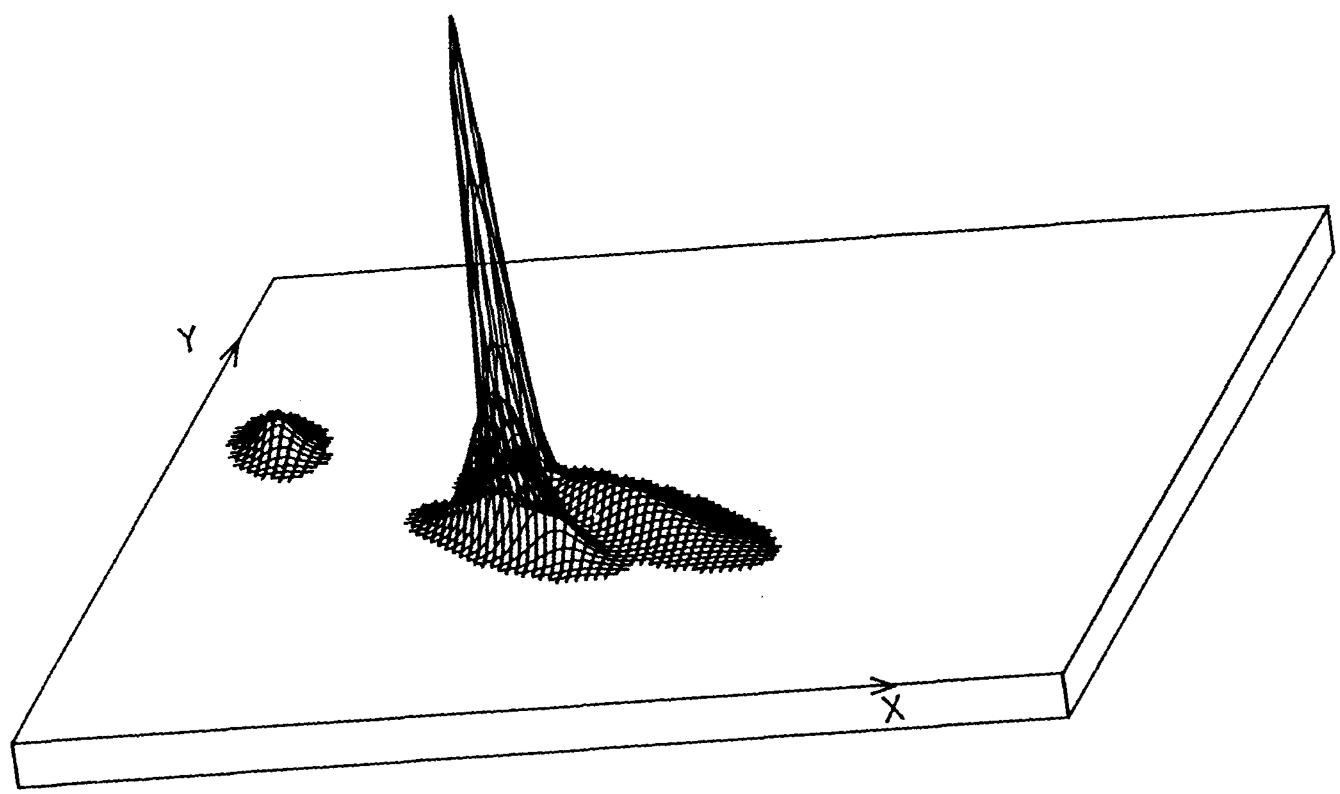


Fig 3.7 Approximated mixture pdf produced by the Joining Algorithm with $N_T = 4$

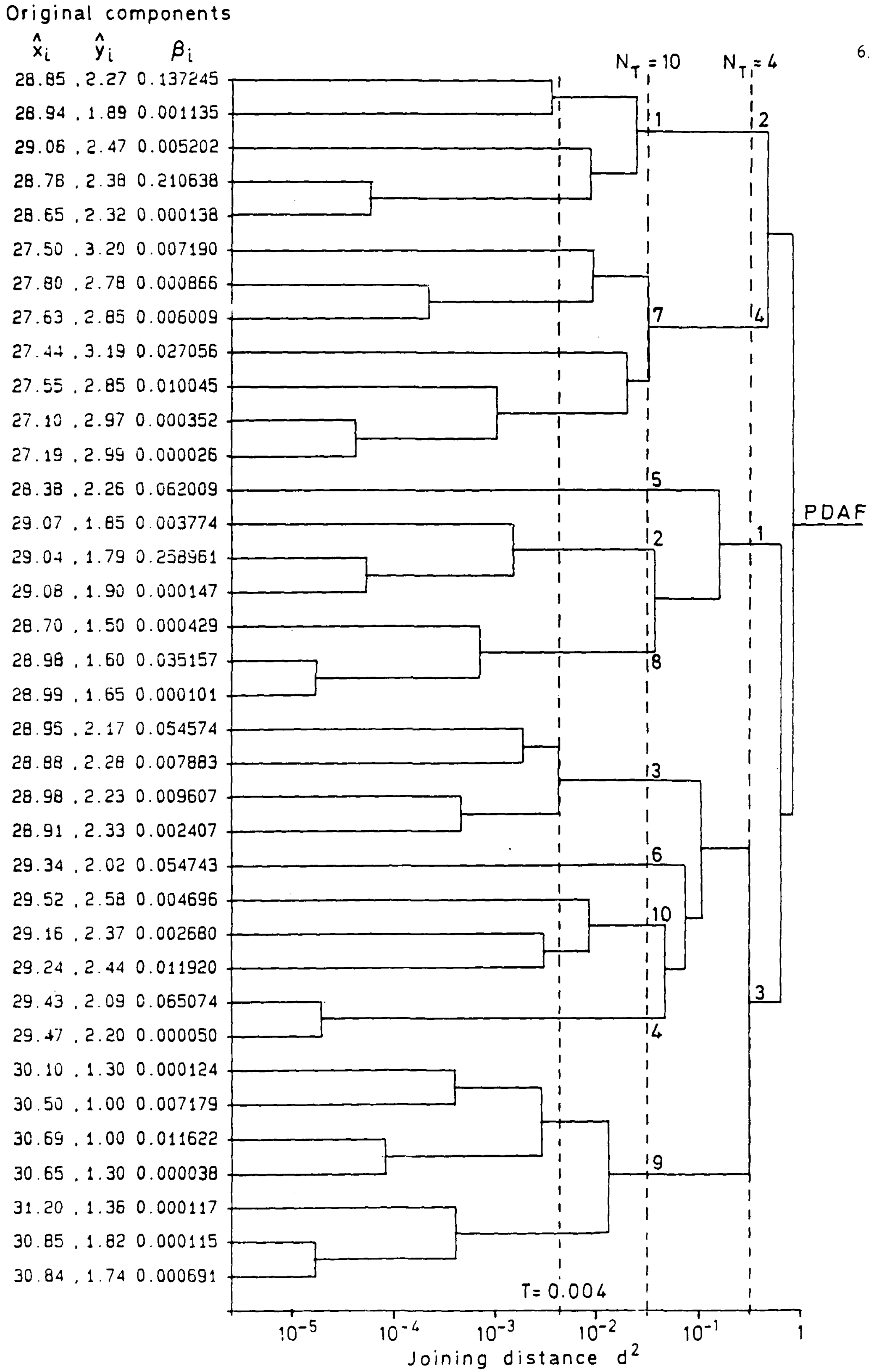


Fig 3.8 History of component merging

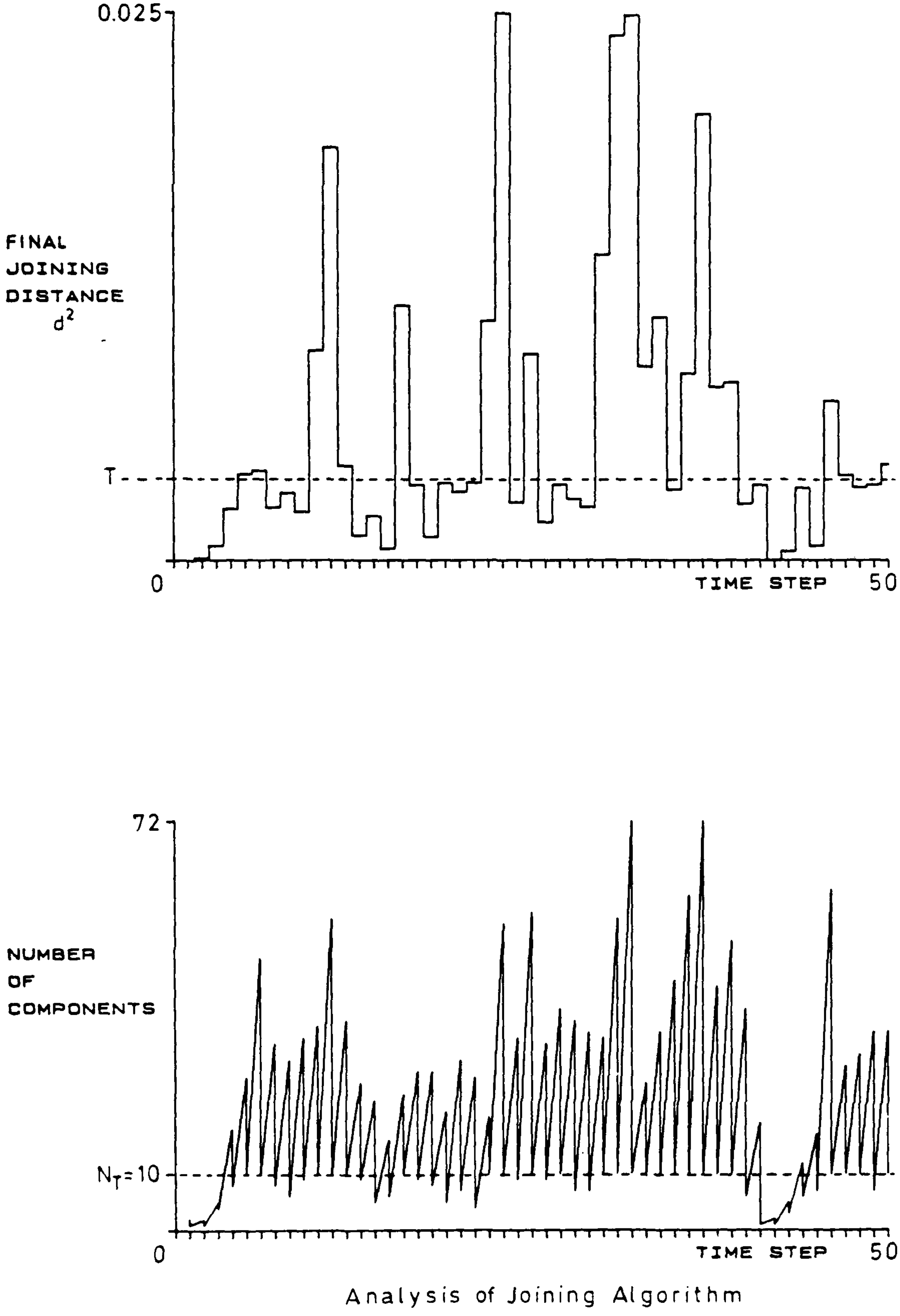
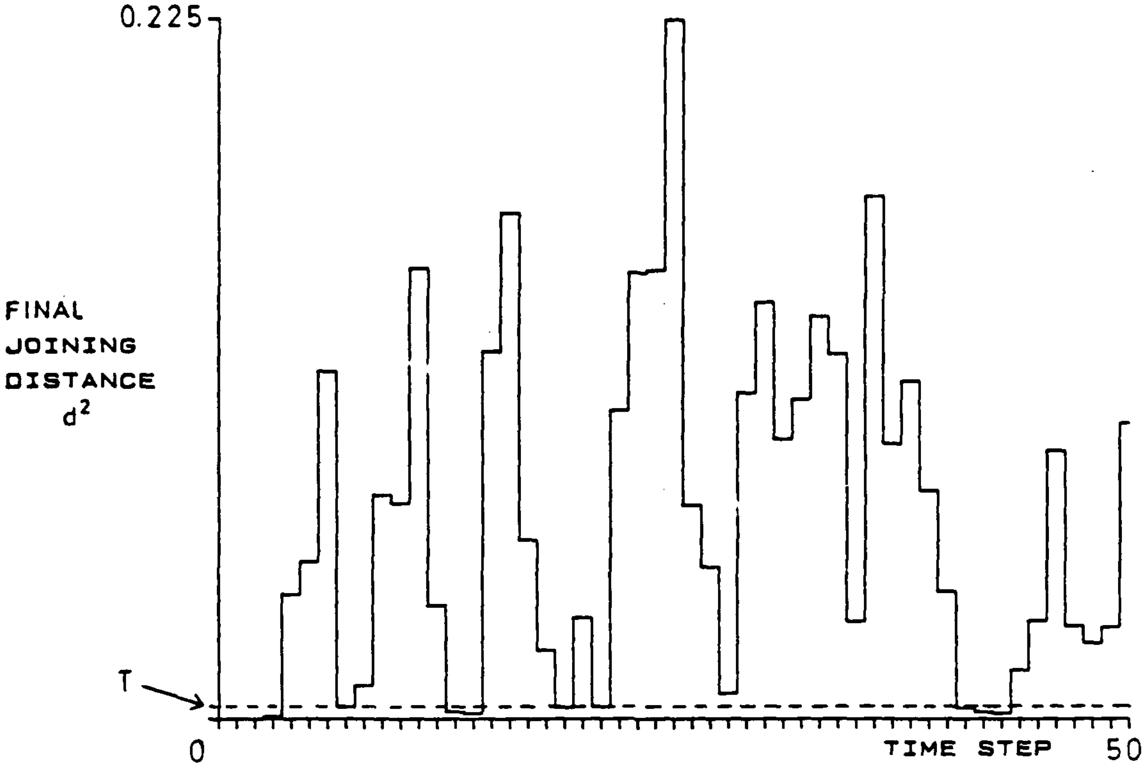
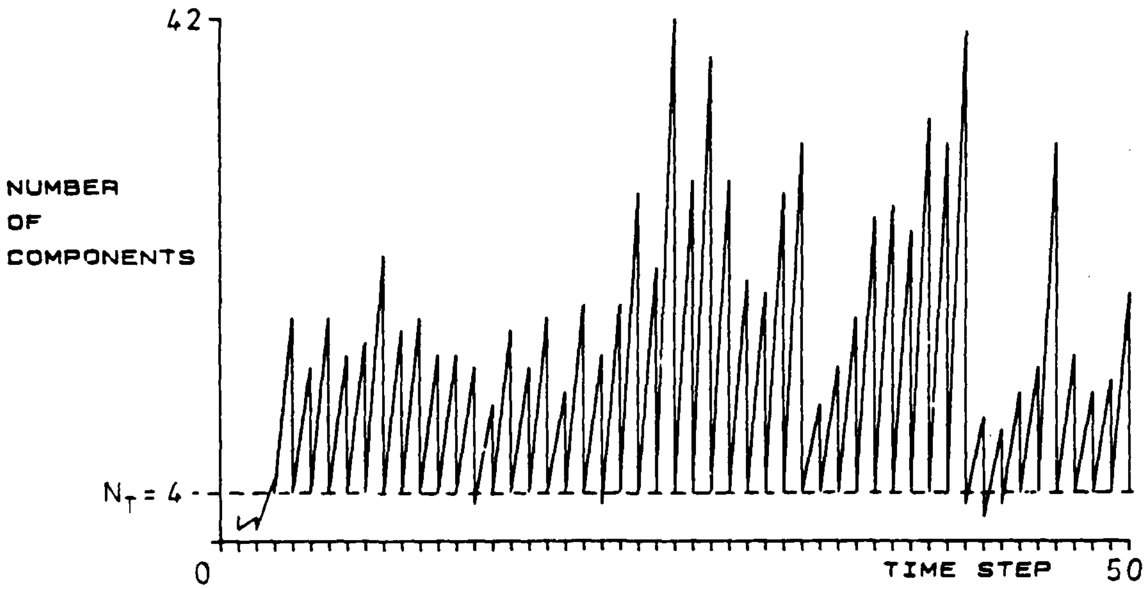


Fig 3.9 Control of mixture components for a tracking example $N_T = 10$





Analysis of Joining Algorithm

Fig 3.10 Control of mixture components for a tracking example $N_T = 4$

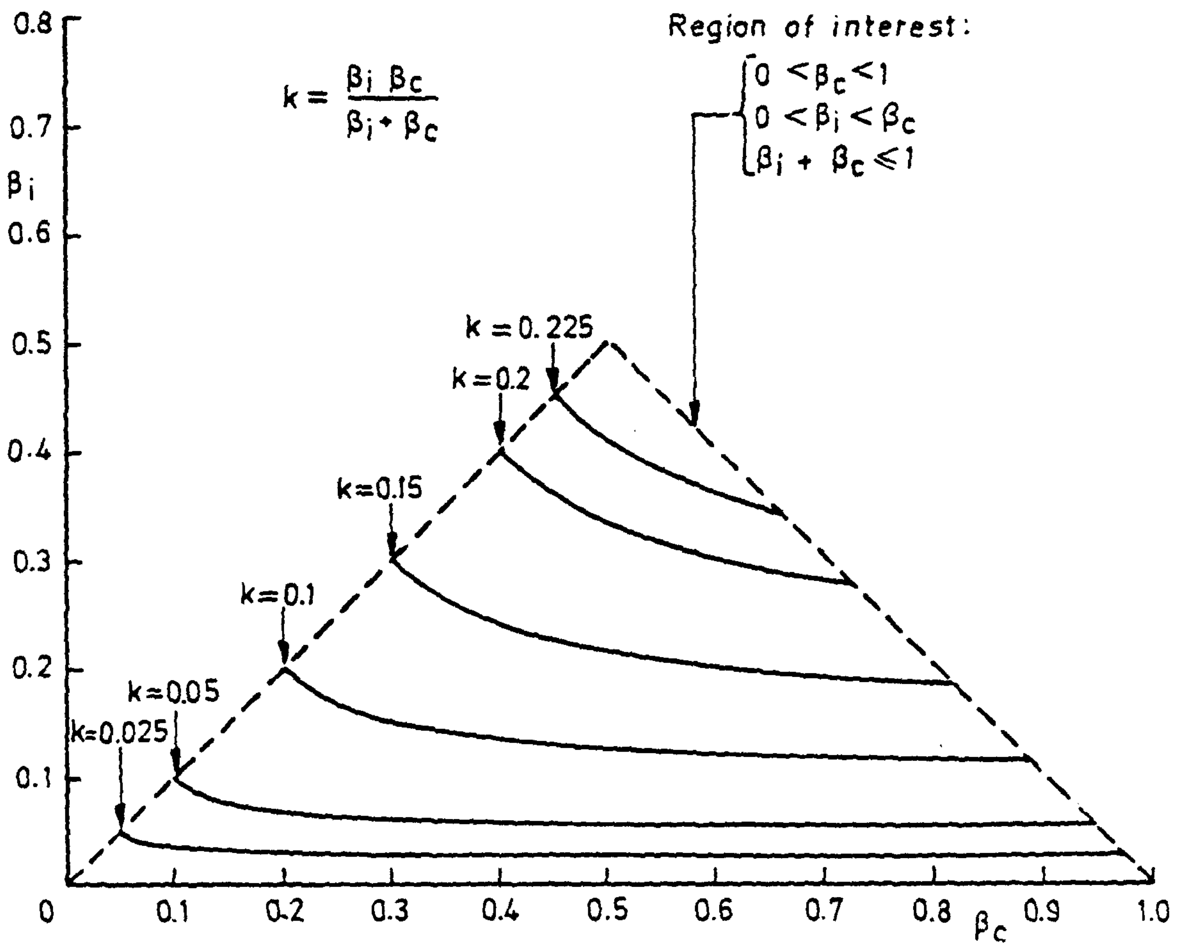


Fig 3.11 Contours of the modifying factor $\beta_i\beta_c/(\beta_i + \beta_c)$

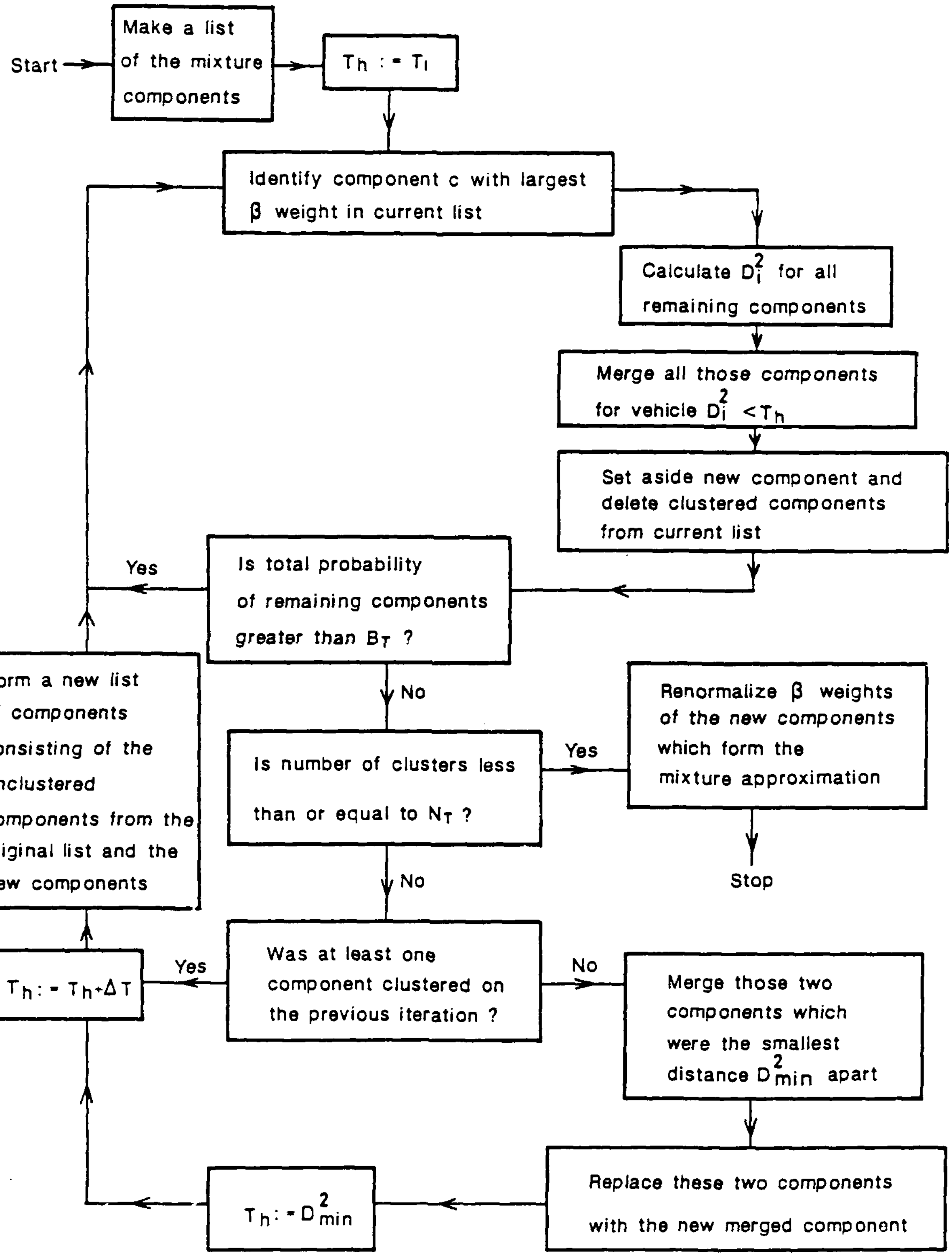


Fig 3.12 Flow diagram of the Clustering Algorithm

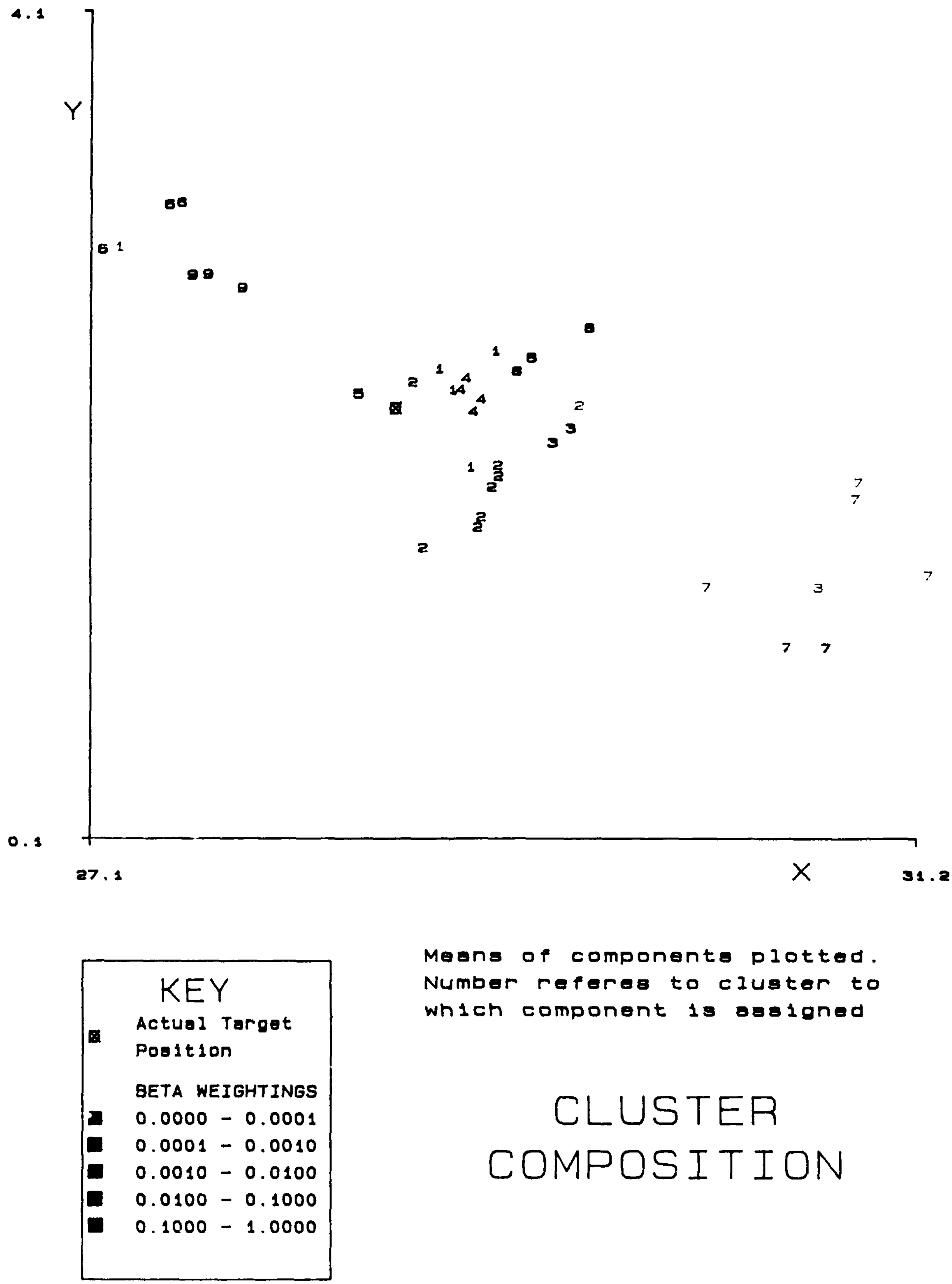


Fig 3.13 Partition of original components into 9 clusters
(Clustering Algorithm with $N_T = 10$)

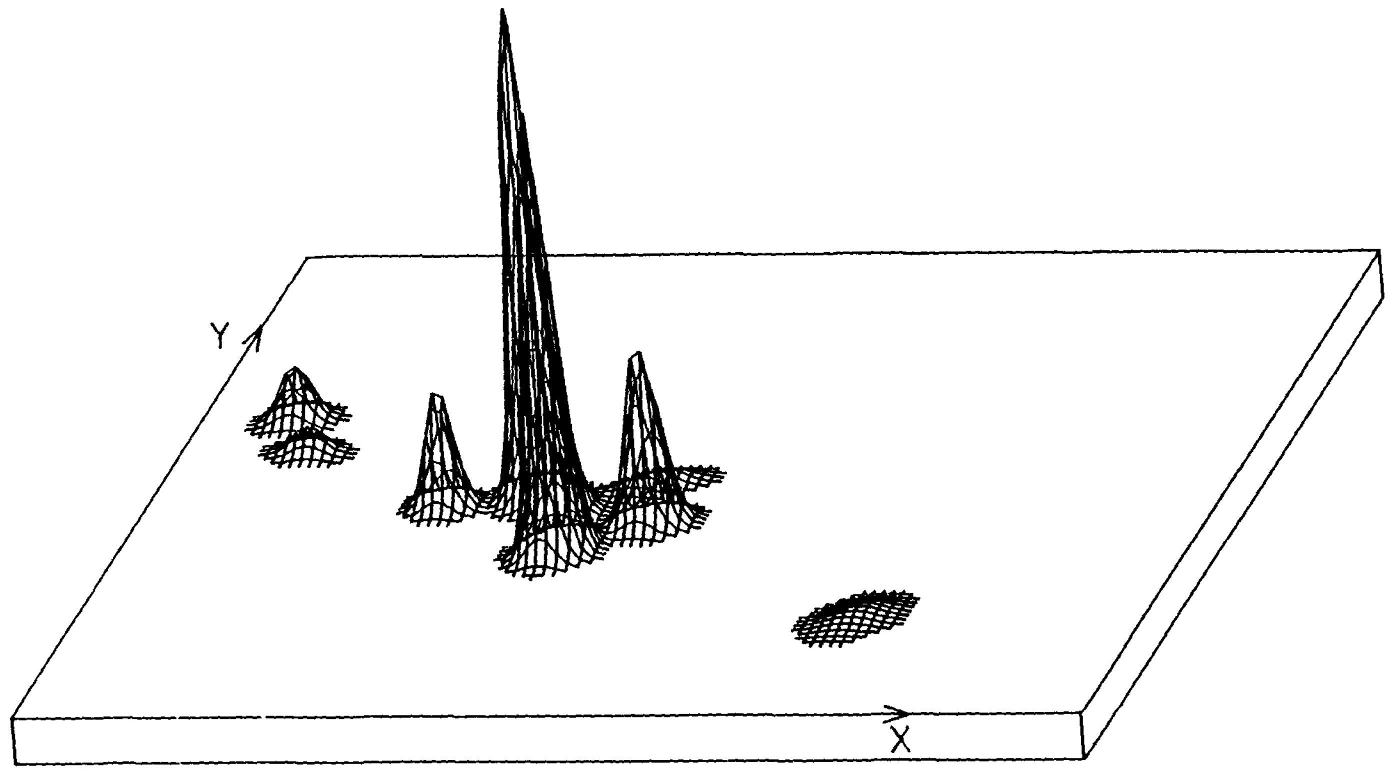


Fig 3.14 The approximated mixture pdf produced by the Clustering Algorithm with $N_T = 10$ (9 components)

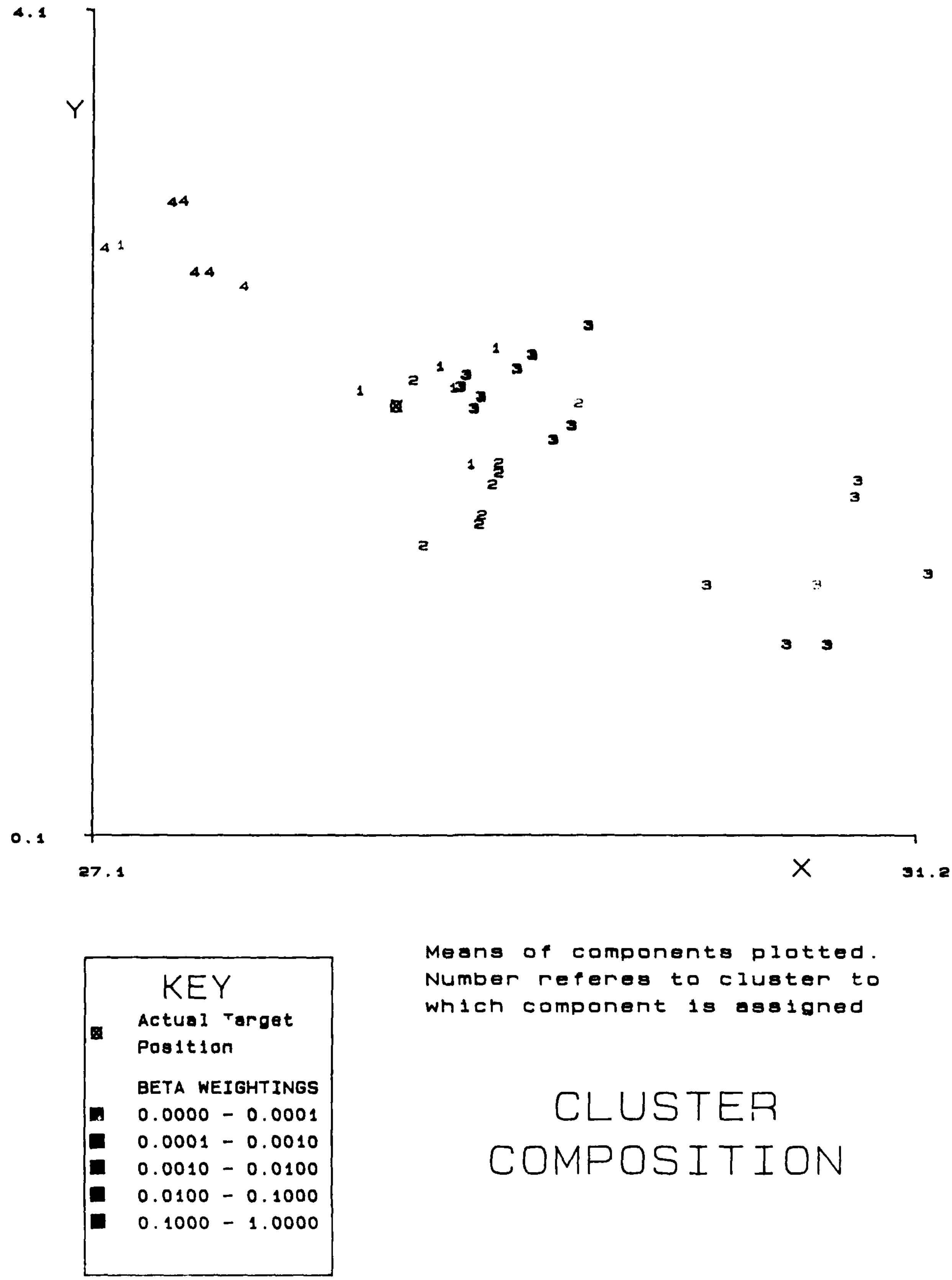


Fig 3.15 Partition of original components into 4 clusters (Clustering Algorithm with $N_T = 4$)

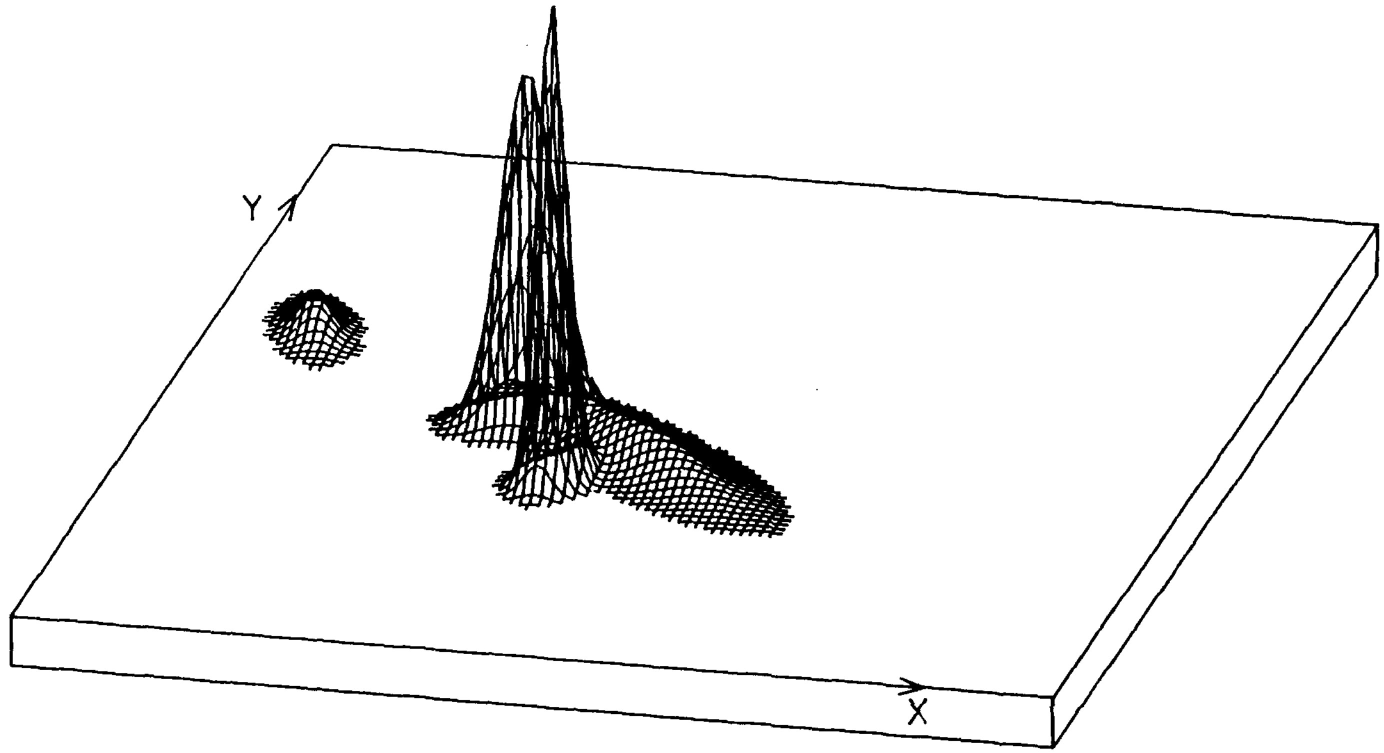
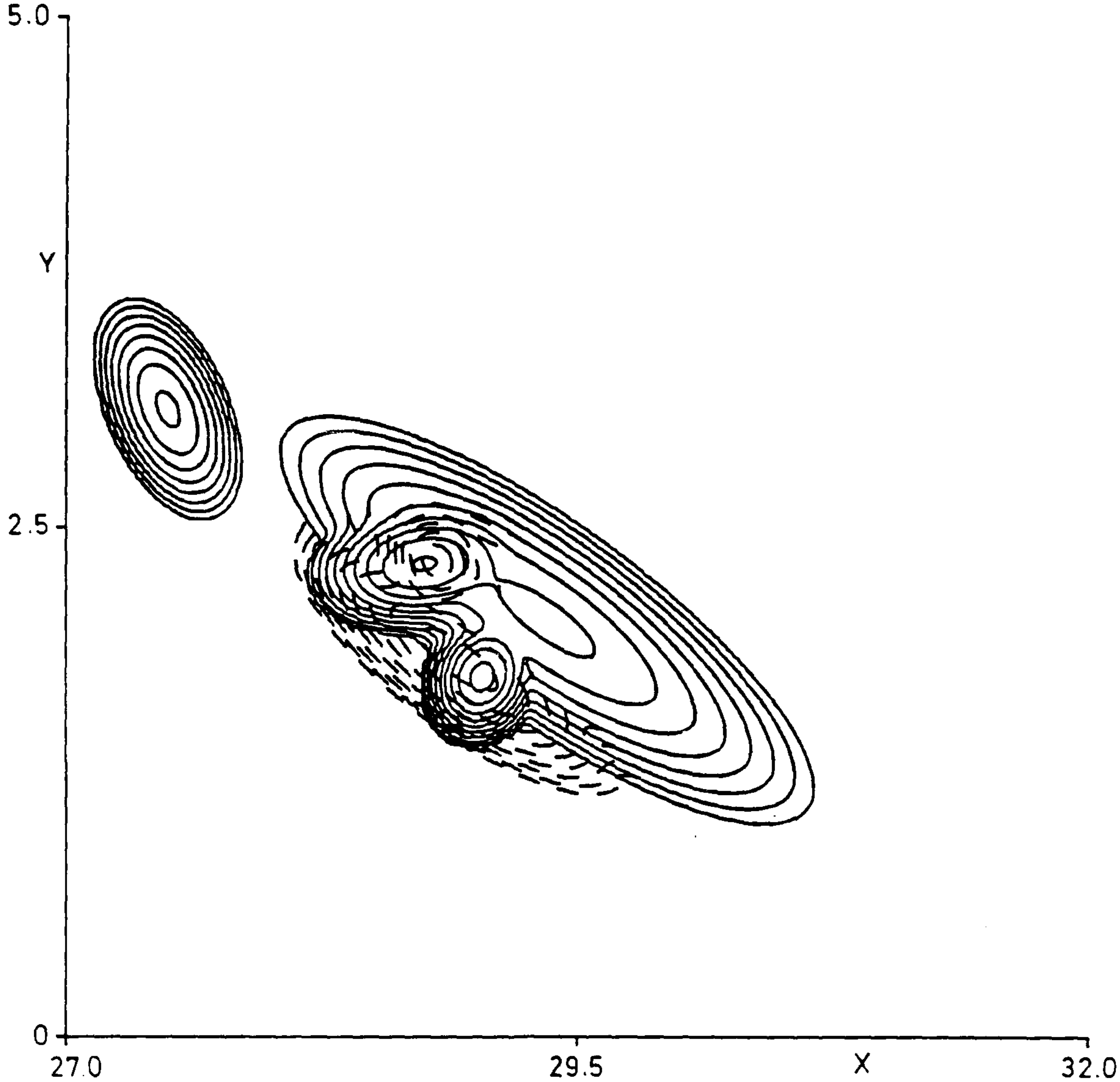


Fig 3.16 The approximated mixture pdf produced by the Clustering Algorithm with $N_T = 4$ (4 components)



Logarithmic contour spacing
Contour heights are :

0.000100	0.000316	0.001000	0.003162	0.010000
0.031623	0.100000	0.316228	1.000000	3.162278

----- ---- Joining Algorithm
———— Clustering Algorithm

Fig 3.17 Comparison of approximated mixture pdf s produced by the Joining Algorithm and the Clustering Algorithm with $N_T = 4$

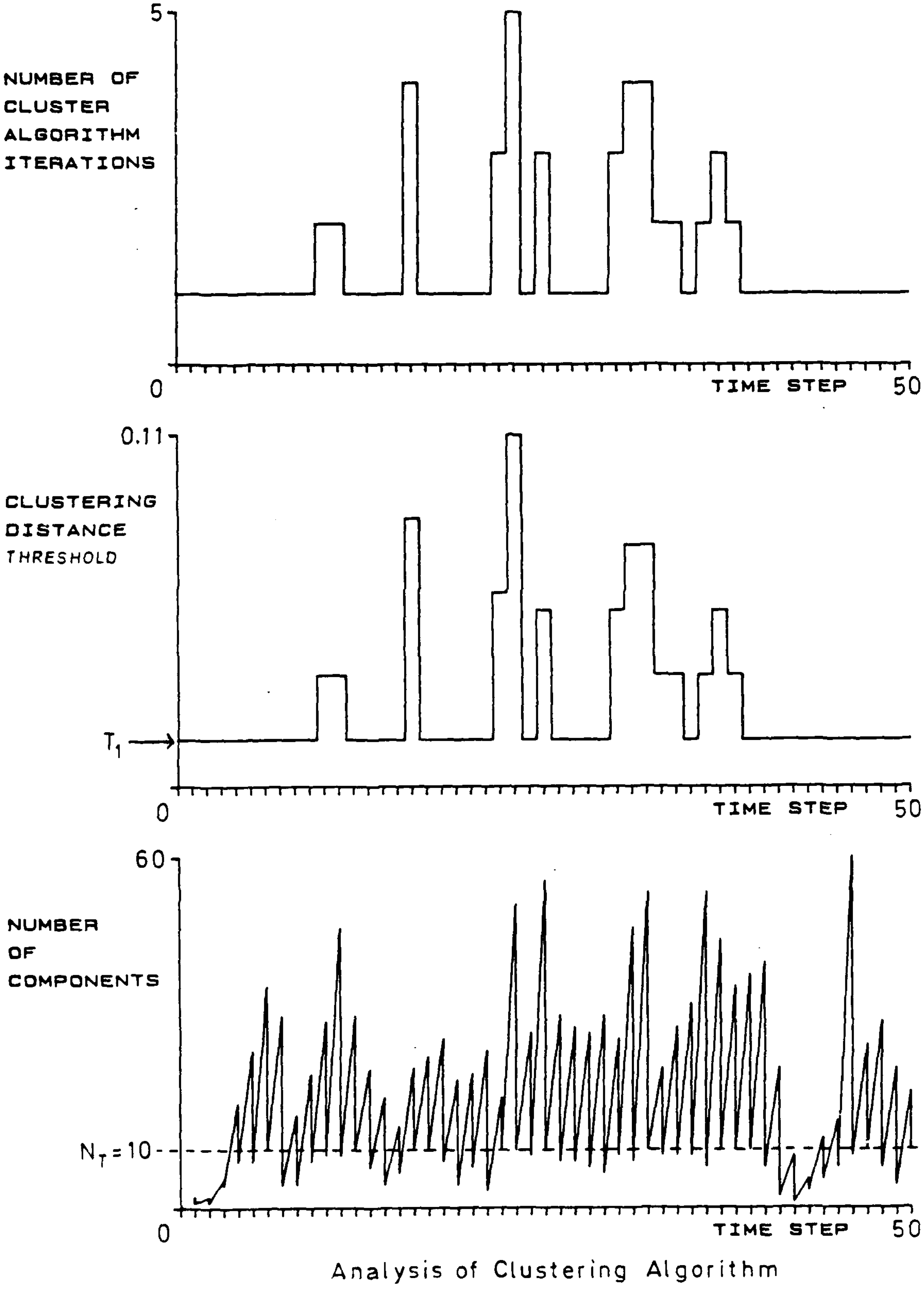


Fig 3.18 Control of mixture components for a tracking example $N_T = 10$

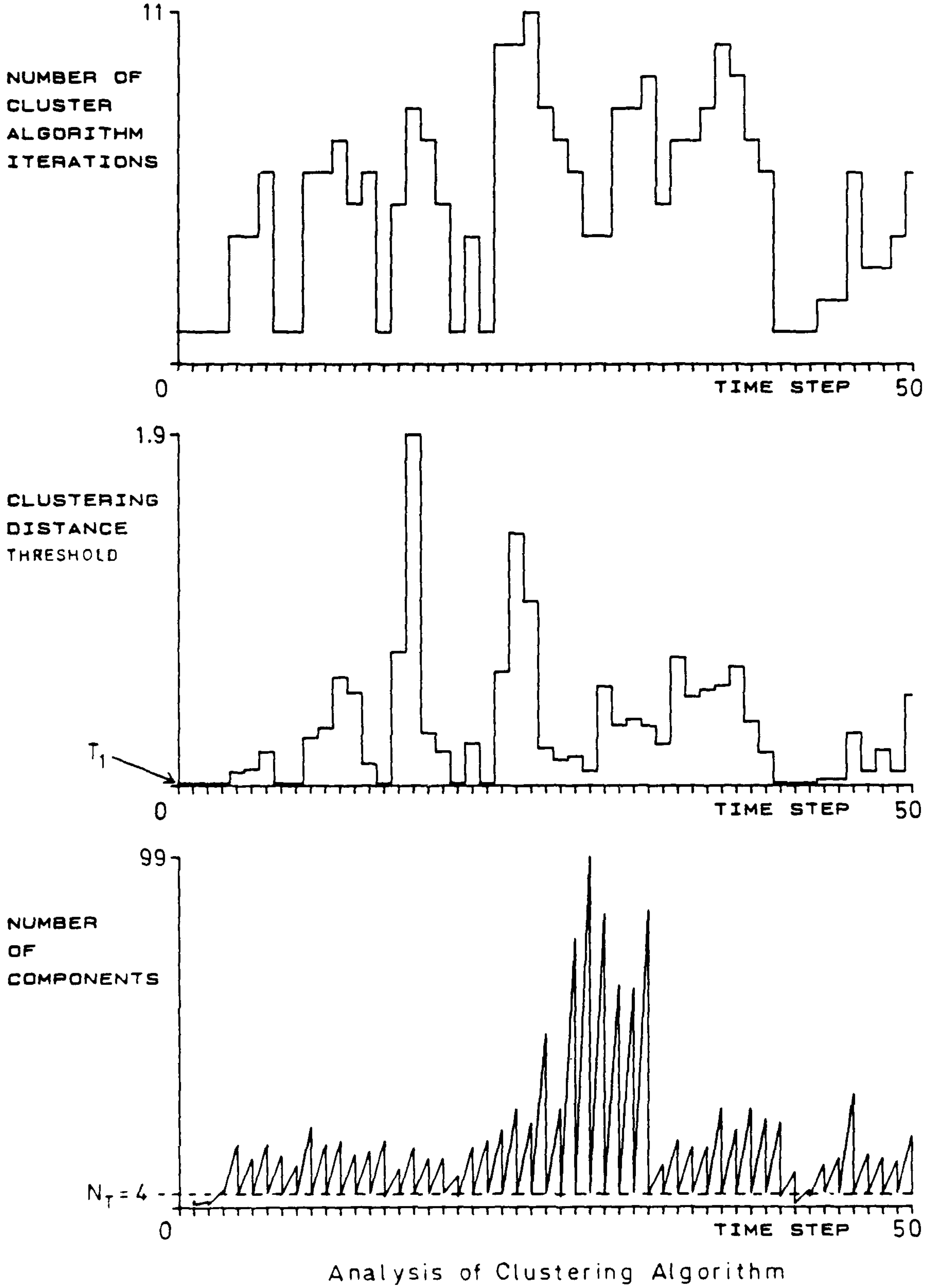


Fig 3.19 Control of mixture components for a tracking example $N_T = 4$

4 PERFORMANCE COMPARISON OF THE JAF WITH THE CAF AND THE EFFECT OF VARYING N_T

4.1 Introduction

Simulation studies are essential for assessing the performance of tracking filters employing the mixture reduction algorithms described in the previous chapter. Since tracking is a statistical operation it is necessary to carry out Monte Carlo simulation runs to obtain estimates of filter performance. Performance has been assessed for an example of the baseline problem: the tracking of a target moving in a plane. The Bayesian solution of Chapter 2 has been programmed for the example, and the approximation techniques of Chapter 3 have been included to produce a Joining Algorithm Filter (JAF) and a Clustering Algorithm Filter (CAF). These filters both employ a coarse acceptance test (see section 3.2) and, save for the reduction technique, they are identical. Also for comparison the single Gaussian approximation PDAF has been programmed.

The main objective of the simulations in this chapter is to compare the performance of the filters and to examine the effect of varying the maximum number N_T of components allowed in the approximation. This has been examined for a single set of problem parameters, chosen at a point in the space where the JAF and the CAF outperform the PDAF. The variation of performance over the problem parameter space for fixed reduction algorithm parameters is assessed in Chapter 5. In all of these simulations, the generated target trajectories and the statistics of the simulated measurements are perfectly matched to the filter parameters. Clearly in real life this is unlikely to be the case. In Chapter 6, filter performance for data statistics mismatched to filter parameters is assessed for a similar tracking problem. Also tracking performance against some 'realistic' trajectories is investigated.

4.2 The tracking problem

Target trajectories have been simulated using a second order model which is the basis of the α-β filter[49,50]. This model has been widely used in tracking problems as it is simple, while providing an adequate trajectory representation for many practical cases. The trajectory described by the model is a variation about a constant velocity course, whose magnitude and direction are defined by initial conditions. The deviation from this mean course is controlled by the variance q of the model driving noise. The second order model is defined by the following equation:

$$\underline{x}_{k+1} = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix} \underline{x}_k + \begin{pmatrix} \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} \\ 0 & \Delta t \end{pmatrix} \underline{w}_k \qquad (4.1)$$

where the state vector $\underline{x}_k$ represents the position and velocity of the target at time $k\Delta t$:

$$\underline{x}_k = (x, \dot{x}, y, \dot{y})_k^T ,$$

Δt is the time step between measurements, and $\underline{w}_k$ is a 2×1 vector from a Gaussian random sequence with zero mean and constant covariance:

$$Q = \begin{pmatrix} q & 0 \\ 0 & q \end{pmatrix} .$$

Thus, to generate a trajectory $\underline{x}_k$, Gaussian random numbers of variance q were fed through the recurrence relation (4.1), starting from some initial condition $\underline{x}_1$. Note that the target velocity described by equation (4.1) is a random walk.

At each time step k , a set of Cartesian position measurements have been generated to simulate sensor measurements. This set consists of at most one true measurement plus uniformly distributed false measurements. The probability of a true measurement occurring is the detection probability P_D . A true measurement $\underline{z}_{kj}$ is a Gaussian perturbation about the target position and it is generated from the state vector $\underline{x}_k$ using the equation:

$$\underline{z}_{kj} = \begin{pmatrix} x \\ y \end{pmatrix}_k + \underline{v}_k \; , \tag{4.2}$$

where $\underline{v}_k$ is a 2×1 vector of Gaussian measurement noise with zero mean and constant covariance:

$$R = \begin{pmatrix} r & 0 \\ 0 & r \end{pmatrix} .$$

The false measurements are independent of the target and are uniformly distributed over the sensor surveillance region, with density ρ per unit area. At each time step, the surveillance region of the sensor is arranged to be sufficiently extensive to include the target position and the acceptance regions of the filters, while track is maintained. False measurements were simulated by generating $A_k\rho$ pairs of uniformly

distributed random numbers with appropriate scaling; A_k being the area of the surveillance region at time step k .

At each time step, every simulated measurement is passed to the tracking filters which attempt to estimate the current target state vector. The following information is available to the filters:

(i) the value of the initial state vector $\underline{x}_1$, so the initial position and velocity of the target is known perfectly,

(ii) the model of target motion, equation (4.1),

(iii) the relationship between the state vector and the true measurement, equation (4.2),

(iv) the statistics of the false measurements (density ρ), the true measurement noise (variance r), and the model driving noise (variance q),

(v) the detection probability P_D of the sensor.

The tracking filters do not know:

(a) the values of the state vector $\underline{x}_k$, or the noise vectors $\underline{v}_k$ and $\underline{w}_k$ at each time step,

(b) the identity of the true measurement.

Clearly this is an example of the tracking problem given in section 2.2 and so the Bayesian solution of Chapter 2 may be directly applied.

4.3 Parameters of the problem

To analyse this tracking problem it is convenient to normalize the variables so that the unit of time is Δt and the unit of distance is $\sqrt{r}$. *Then the non-dimensional form of the state vector is:*

$$\underline{x}_k' = \left(\frac{x}{\sqrt{r}} , \frac{\dot{x}\Delta t}{\sqrt{r}} , \frac{y}{\sqrt{r}} , \frac{\dot{y}\Delta t}{\sqrt{r}} \right)_k^T .$$

If the target model and measurement equations are written in the normalized form, it can be shown that the statistics of the problem are completely specified by three non-dimensional parameters:

(i) $$\frac{q\Delta t^4}{r} ,$$

the ratio which determines the values of the filter gains for the standard α–β filter, *ie* in the absence of false measurements. As this parameter increases the α–β filter becomes more responsive to position measurements.

(ii) ρr , the expected number of false measurements falling within a square whose side is one standard deviation of the measurement error.

(iii) P_D , *the detection probability.*

Since the initial state vector is assumed to be known perfectly, the filter performance in normalized co-ordinates should only depend on these three parameters. (This is because the problem may be written as the estimation of the deviation about the nominal constantly velocity course defined by the initial state vector.)

The filter performance comparisons reported in the chapter are for a single point in the parameter space:

$$\frac{q\Delta t^4}{r} = 1$$

$$\rho r = 0.012$$

and

$$P_D = 1 \ .$$

These values have been chosen to illustrate the possible improvement in tracking performance of the new reduction algorithms over the PDAF. A full investigation of filter performance over the parameter space is reported in Chapter 5. For the above parameters, the equivalent Kalman filter (receiving only true measurements) rapidly reaches steady state conditions, and the standard deviation of the position error on one of the co-ordinates approaches within 1% of its final steady state value after only four time steps. Also the expected number of false measurements that would be received by an acceptance gate with $P_M = 0.001$ based on steady state Kalman filter covariances is 2.084 (see section 3.2). In the simulations, the initial target position was taken as the origin, the initial speed was $10\sqrt{r}/\Delta t$ and the initial heading was chosen randomly from a uniform distribution over $[0,2\pi]$ for each replication. As noted, initial target position and velocity do not affect the filter performance.

4.4 Track loss criterion and simulation program

The performance of the filters was assessed by measuring how long they were able to maintain track on the target, *ie* the track lifetime.

Each filter was allowed to continue tracking the target until track was lost. A track was deemed to be lost if either of the following criteria were satisfied:

(i) The true measurement is rejected by the acceptance test for five consecutive time steps.

(ii)
$$|\hat{x}_k - x_k| > 10\ \sigma_{xk}$$

or

$$|\hat{y}_k - y_k| > 10\ \sigma_{yk}$$

for five consecutive time steps, where $(\hat{x}_k , \hat{y}_k)$ is the filter estimate (the mean of the posterior distribution) of the target position at time step k , (x_k , y_k) is the actual target position at time step k , and σ_{xk} and σ_{yk} are the standard deviations of the position estimates of the equivalent Kalman filter (*ie* the optimal filter for the same problem but with $\rho = 0$).

These track loss criteria are testing for consistent rejection of the true measurement, or a tracking error which is consistently large in comparison with the expected error of the equivalent Kalman filter; consistent being defined as five time steps and large being defined as ten standard deviations.

One hundred target trajectories with associated measurements were generated, so that the mean track lifetime and the distribution of lifetimes could be estimated. The same hundred trajectories and measurement sets were used for each filter at each setting of N_T , which was varied between 1 and 30.

In practice, to avoid storing and reading large amounts of data, the trajectory and measurements were generated as they were required by the filter at each time step; *ie* data generation and filtering were performed within a single computer program. The track loss test and other assessment operations were also performed within this program, which was used for the CAF/JAF comparison of this chapter and to produce results for Chapter 5. The program includes two tracking filters: the Bayesian filter of Chapter 2 and the PDAF which provides a useful baseline for comparison. The Bayesian filter may be run with either the Joining Algorithm to give the JAF or with the Clustering Algorithm to give the CAF.

All computer programs were written in Fortran 77 and the filter simulations were run on the Cray 1S at RAE Farnborough. Thus where cpu times are quoted, they are for this Cray computer. Due to the structure of the algorithms, the 'vector' processing capabilities of the Cray were hardly used.

4.5 Results

4.5.1 Average number of time steps to track loss

Fig 4.1 shows the average number N_{AVE} of time steps until track loss as a function of N_T, for filters using the Clustering Algorithm and the Joining Algorithm with thresholds set to the values given in sections 3.6.1 and 3.7.1. $N_T = 1$ corresponds to the special case of the PDAF, and clearly the filters which retain more than one mixture component perform better than the PDAF for this example. The Joining Algorithm filter gives slightly larger values of N_{AVE} than the Clustering Algorithm, possibly due to the setting of the thresholds T and T_1.

Also shown in Fig 4.1 is the filter performance for the JAF with $T = 0$, *ie* with the acceptable modification check switched off. Note that the original setting of T for the JAF does not significantly degrade the filter's performance, and the the performance for all three cases shown in Fig 4.1 is similar. For $N_T < 10$, N_{AVE} rises approximately linearly with N_T , while for $N_T > 10$, N_{AVE} is nearly constant. Thus, for this example, $N_T = 10$ appears to be about the critical level below which tracking performance begins to degrade. (The mechanism of track estimation is discussed in Chapter 5.) For the JAF with $T = 0$ and N_T very large, the mixture is not subject to approximation, and so this constant level is the optimal value of N_{AVE} .

Fig 4.2 shows the average number of mixture components before and after reduction for the three cases of Fig 4.1. Comparing Fig 4.2a&b with 4.2c, the effect of the acceptable modification check, defined by T_1 or T , in regulating the number of components for the large values of N_T is obvious. For small values of N_T , the approximation for all three cases is principally controlled by N_T itself. For this example, T_1 and T become the main regulators of the approximation at about $N_T = 10$, so the acceptable modification check appears to select the minimum nu of components for near optimal performance. Clearly this cannot be guaranteed for other tracking problems, but since the thresholds were not specially tuned for this simulation, the performance with other problems may not be far from optimal.

4.5.2 <u>Distribution of number of time steps to track loss</u>

In the previous section, the average track lifetime was discussed. In this section we consider the distribution of track lifetimes about

this mean. To illustrate the distribution and to compare the performance of the CAF and JAF for individual replications, the track maintainance times have been plotted in Figs 4.3, 4.4 and 4.5 for $N_T = 2, 4$ and 30 respectively. In these diagrams each point corresponds to a single replication, and the X and Y co-ordinates of the point are the time steps at which the JAF and CAF (with original threshold settings) lost track respectively. So points falling on the $X = Y$ line indicate that both filters lost track coincidently. For large values of N_T (*eg* $N_T = 30$, Fig 4.5), the performance of the two filters is remarkably similar for the majority of replications. The few replications biasing N_{AVE} in favour of the JAF are obvious. For small values of N_T (*eg* $N_T = 2$, Fig 4.3), the points are scattered further from $X = Y$ although N_{AVE} is almost identical for the two filters. These results bear out the observation that the mixture approximations produced by the two reduction algorithms are usually very similar for large N_T, while for small N_T there are often clear differences.

Figs 4.6 and 4.7 show histograms of the data points from Figs 4.3 and 4.5; *ie* for the track lifetimes for the JAF and CAF with $N_T = 2$ and $N_T = 30$. It can be seen that those track lifetimes exceeding 20 time steps can be well fitted by an exponential distribution of the form:

$$p(t) = \begin{cases} \dfrac{1}{\alpha} e^{-\left(\frac{t-t_{min}}{\alpha}\right)} & \text{for} \quad t > t_{min} \\ 0 & \text{otherwise} \, , \end{cases}$$

where $(t_{min} + \alpha)$ is the average lifetime of tracks which survive for at least $t_{min} = 20$ time steps. This is confirmed by a χ^2 test:

the exponential hypothesis is only once rejected at the 5% level of significance for any of the 24 sets of replications. This exponential distribution indicates that after 20 time steps, the probability of losing track is independent of track lifetime, *ie* after an initial transient the filters reach steady state conditions. The value $t_{min} = 20$ was chosen by examining the transient behaviour of the equivalent Kalman filter (see last paragraph of section 4.3) and by inspection of the simulation results. The distribution parameter α may be interpreted as the average number of time steps that a track will survive in steady state conditions. Estimates of α are shown in Fig 4.8. These values are slightly greater than $N_{AVE} - 20$, as tracks surviving for less than 20 time steps are excluded.

It is important to establish the distribution of track lifetimes as this allows one to specify confidence limits on the estimate of α. For an exponential distribution, the 95% confidence limits are approximately:

$$\left(\alpha \left(1 + \frac{1.96}{2\sqrt{N}} \right)^{-2} , \ \alpha \left(1 - \frac{1.96}{2\sqrt{N}} \right)^{-2} \right) ,$$

where N is the number of replications used to estimate α. These limits define a fixed interval when track lifetime is plotted on a logarithmic scale. In the performance estimates of the following chapters, these confidence limits are shown with N_{AVE}, on the assumption that track survival times are also exponentially distributed in these cases and that t_{min} is small compared with N_{AVE}.

4.5.3 Computation time

Fig 4.9 shows the average cpu time T_{AVE} for the filters to perform a single time step. The time scale (which is logarithmic) is normalized to the average cpu time for a single PDAF time step which, for the data simulated here, was 1.12 ms on a Cray 1S computer. The computational effort is divided between the propagation of mixture components or tracks and mixture reduction. For the two filters with the original threshold settings (Fig 4.9a&b), T_{AVE} falls rapidly to nearly constant values for $N_T > 10$. Also for low values of N_T most time is spent reducing the mixture, and as N_T increases more time is required for track propagation while the mixture reduction time decreases. This is explained by Fig 4.2: the initial high values of T_{AVE} are due to time spent reducing large mixtures which result from inadequate approximations at values of $N_T < 6$. Except for the case $N_T = 6$, the JAF was more time consuming then the CAF, usually by about 50%, and as expected, the execution times for the filters were in all cases considerably greater than the PDAF. However for $N_T > 10$, the five fold increase in execution time for the CAF may well be an acceptable price for the performance improvement offered by this filter.

The time taken by the JAF with $T = 0$ is shown in Fig 4.9c. This clearly shows the value of the acceptable modification check in the reduction algorithms: for the insignificant improvement for $N_T > 10$ over the filter with the original threshold settings, there is a large increase in computational overheads. The extra processing time is required for the propagation and reduction of the extra tracks generated when the full N_T components are retained for $N_T > 10$, (see Fig 4.2).

4.6 Conclusions

For the chosen simulation example, the JAF and CAF both give a substantial performance improvement over the PDAF. The penalty for this is the increased computational requirements of the more complex filters. Minimum computation time and near optimal performance were obtained when satisfactory mixture approximation (defined by the algorithm thresholds T and T_1) was achieved within the maximum number N_T of mixture components allowed. Under these conditions the track survival times for the JAF and CAF were identical on at least 85% of the replications. This suggests that filter performance is not highly sensitive to the method of mixture reduction, provided that the most important mixture components are retained. However, the computation time for the JAF was almost always greater than that for the CAF, usually by about 50%. Thus in the remainder of this study the Clustering Algorithm is always used for mixture reduction.

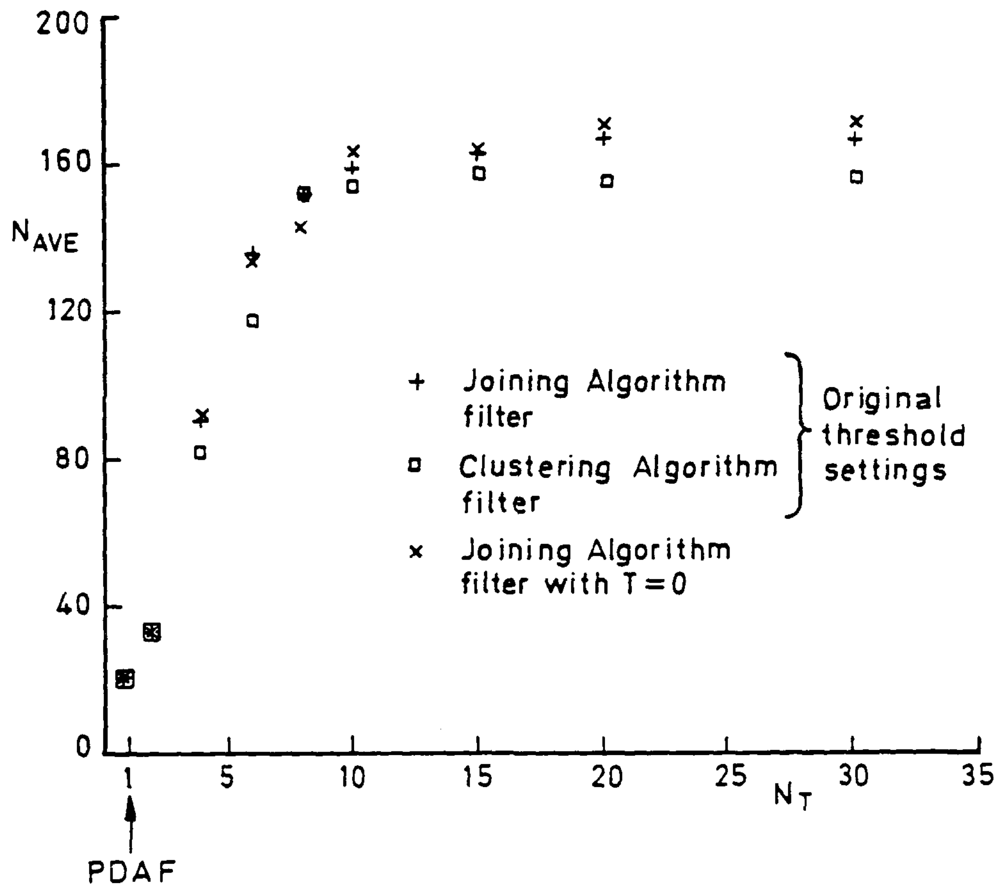


Fig 4.1 The average number of time steps until track loss as a function of N_T

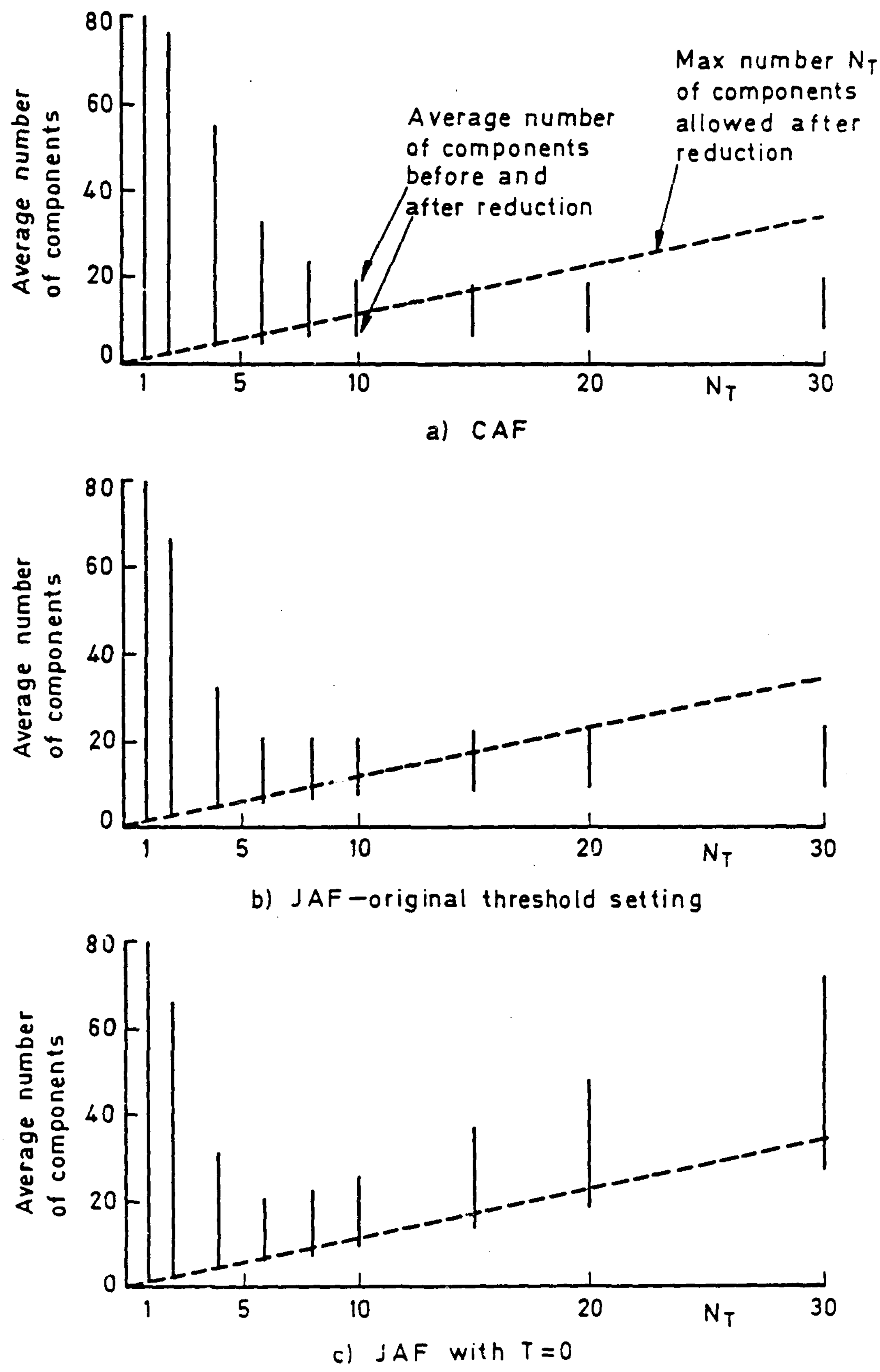


Fig 4.2 Average number of mixture components before and after reduction

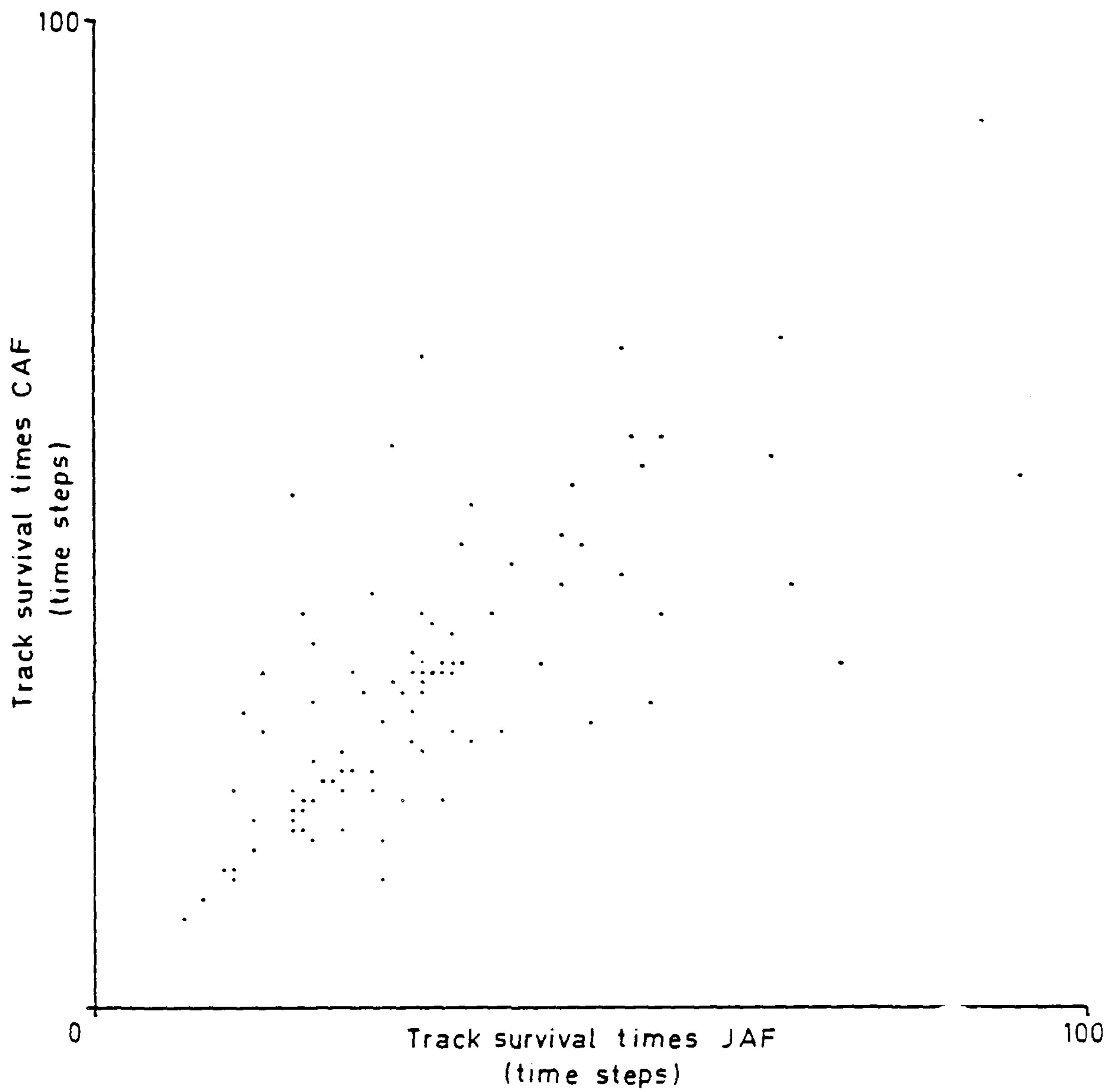


Fig 4.3 Track maintenance times for each replication $N_T = 2$

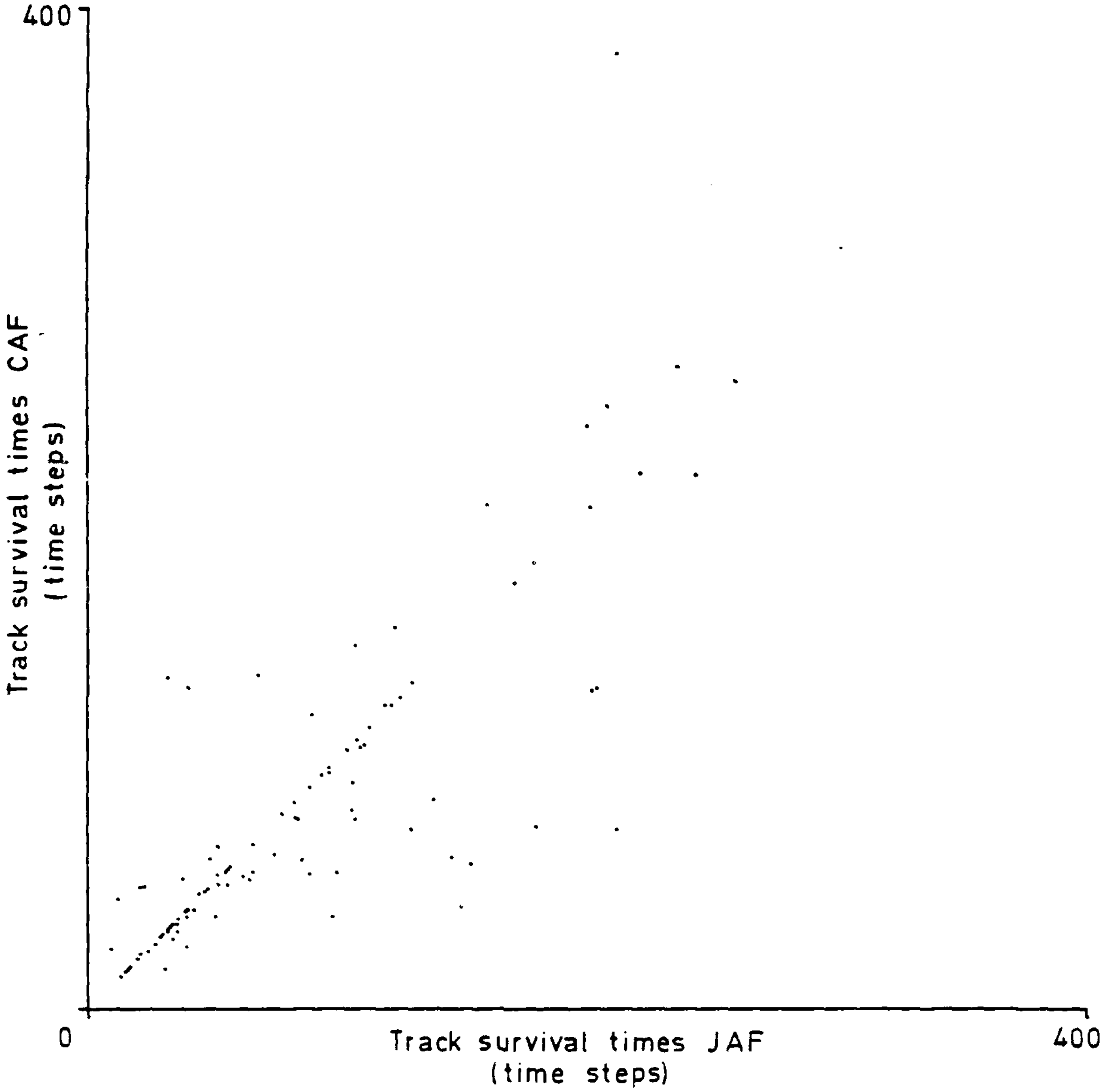


Fig 4.4 Track maintenance times for each replication $N_T = 4$

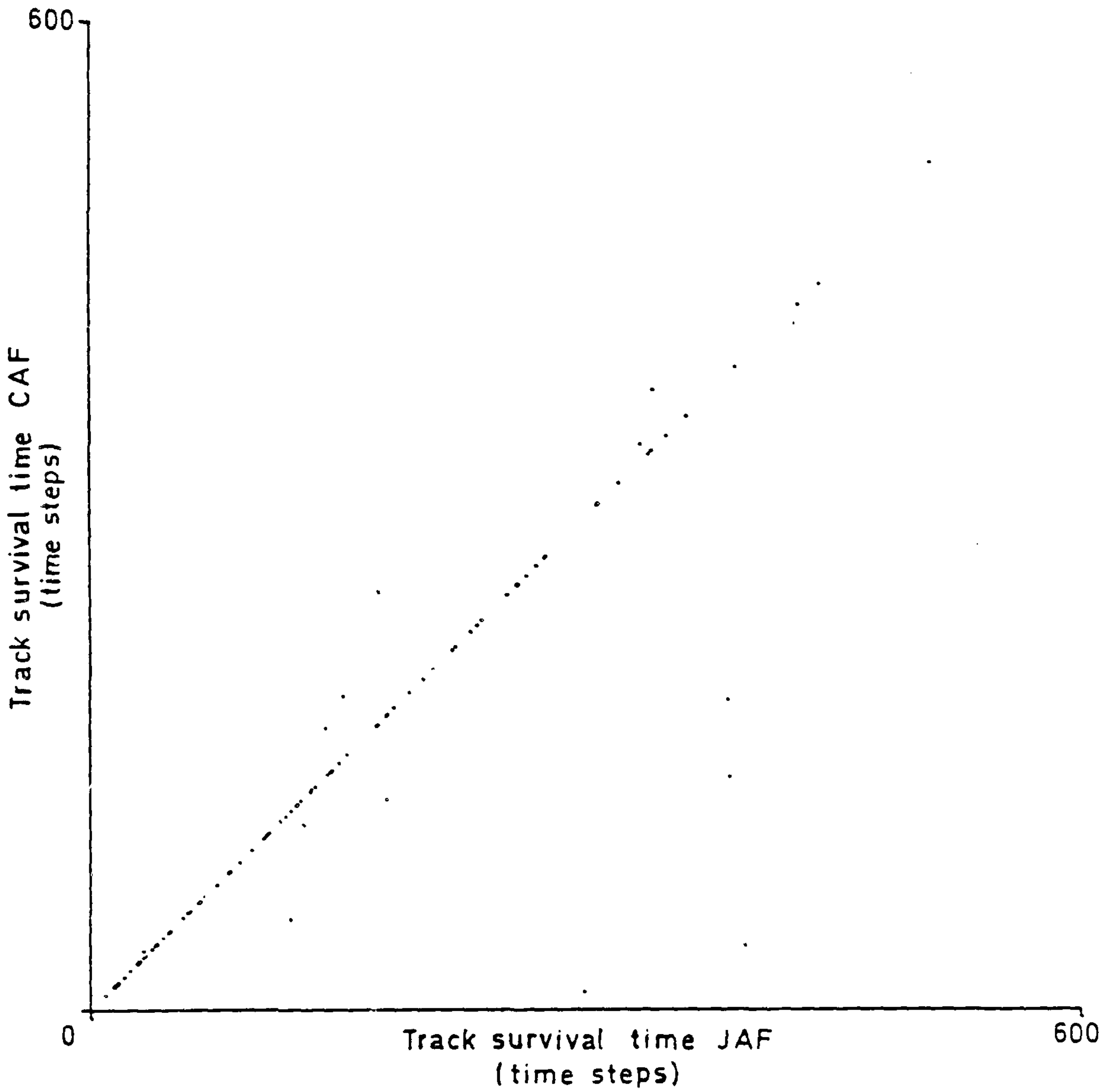
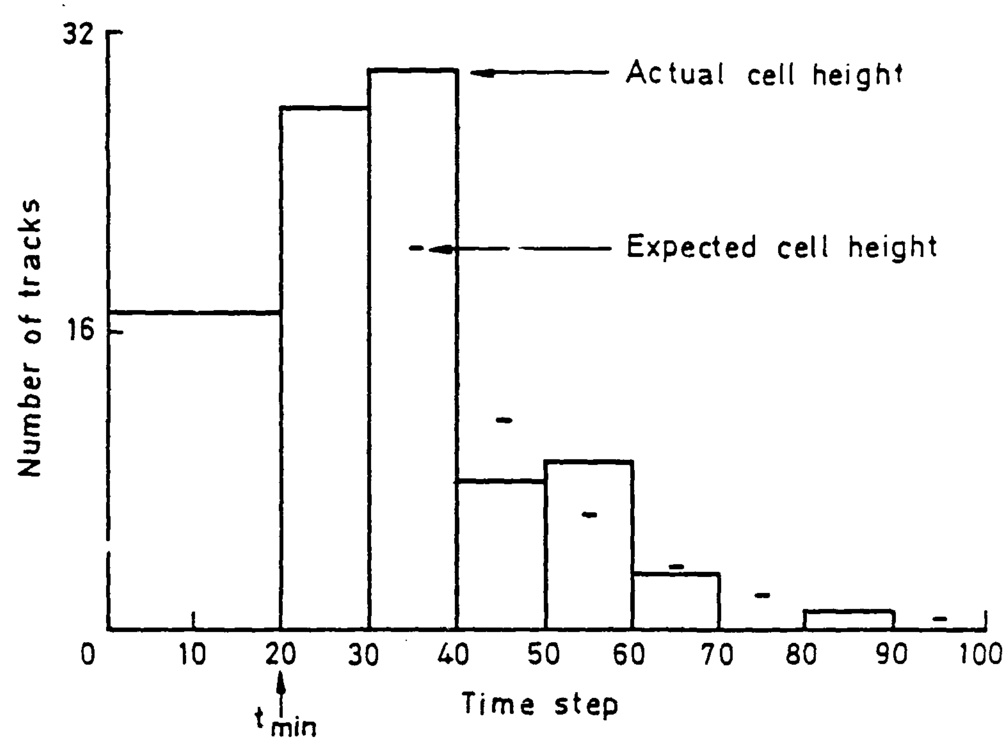
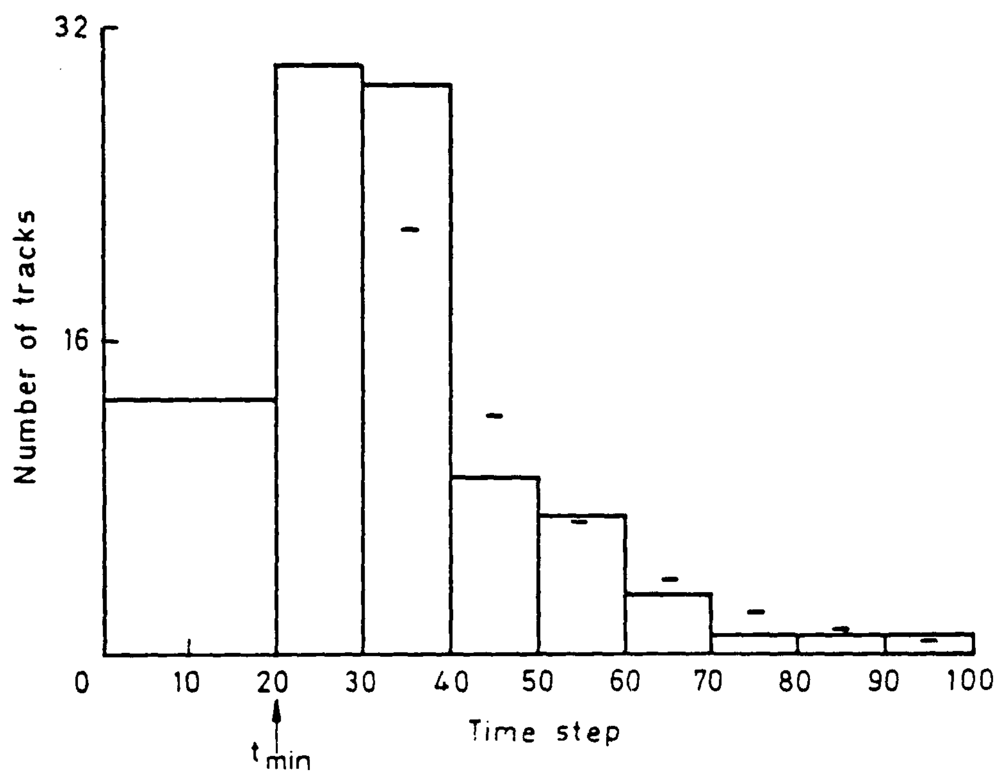


Fig 4.5 Track maintenance times for each replication $N_T = 30$

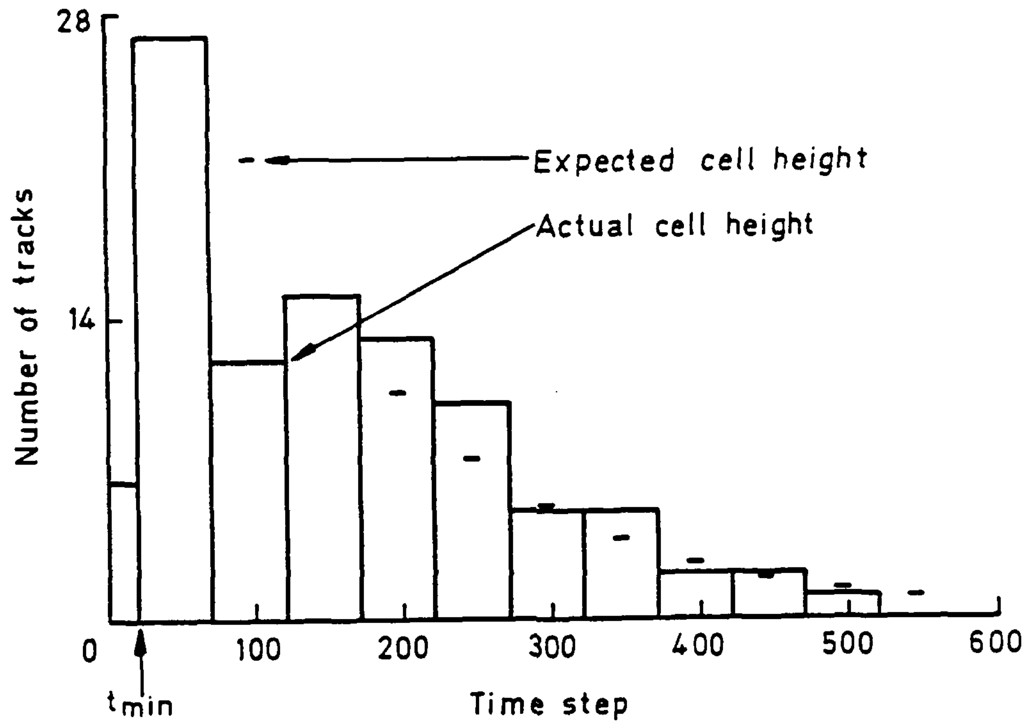


a) CAF $N_T = 2$

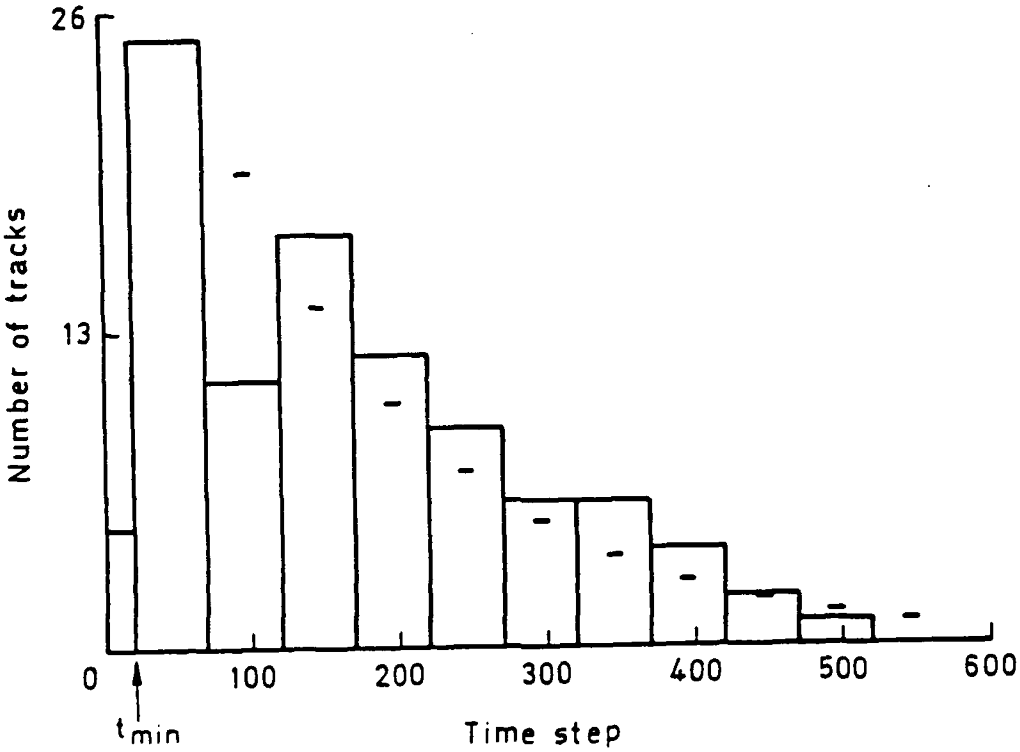


b) JAF $N_T = 2$

Fig 4.6 Histograms of track survival times for N_T = 2



a) CAF $N_T = 30$



b) JAF $N_T = 30$

Fig 4.7 Histograms of track survival times for N_T = 30

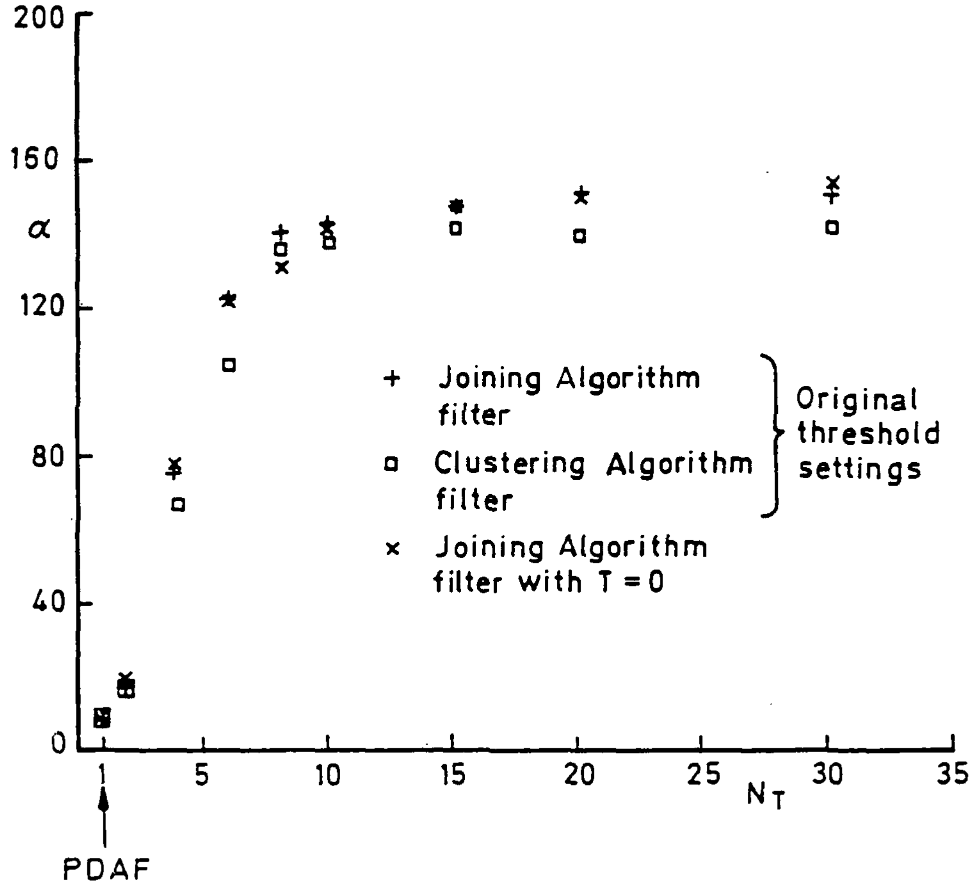


Fig 4.8 Estimates of average track survival times in steady state conditions

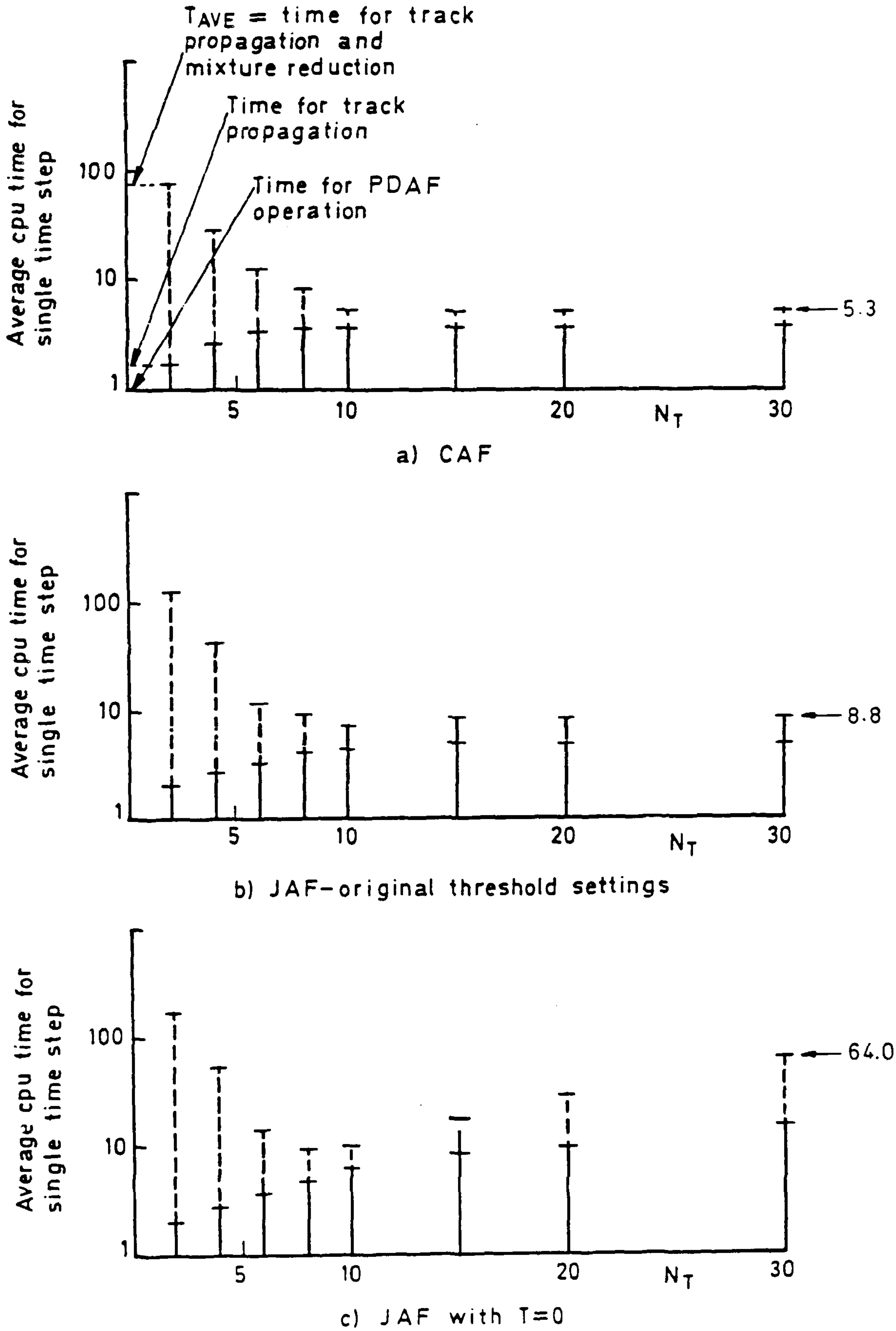


Fig 4.9 Average cpu time to perform a single time step

5 SIMULATION STUDIES OF THE CAF AND THE PDAF

5.1 Introduction

The performance of the tracking filters for the problem described in section 4.2 depends on only three problem parameters. If the probability of detecting the true measurement is unity, the two remaining parameters are ρr, the normalized density of false measurements, and $q\Delta t^4/r$, the normalized acceleration variance of the target. The primary aim of this chapter is to examine the performance of the CAF as a function of these two parameters. The performance measure is the average track survival time N_{AVE}, and the baseline for the assessment is the performance of the PDAF. We shall attempt to identify the region of the parameter space where the more complex CAF gives a significant performance improvement over the PDAF. In the light of the simulation example of the previous chapter, the maximum number N_T of components that may be retained by the Clustering Algorithm has been set at 20. It is hoped that these simulation results will provide an assessment and design aid for this type of tracking problem.

In the second part of this chapter (section 5.3), a single run of the tracking filters is examined in detail. The purpose of this demonstration is to give a physical insight into how the Bayesian filter estimate is produced. The example is of a situation where track loss may be avoided by the retention of more than one mixture component (using the Clustering Algorithm).

5.2 The performance of the CAF over the problem parameter space

5.2.1 Presentation of results

The average track survival time N_{AVE} for the CAF and the PDAF is shown as a function of ρr in Figs 5.1 to 5.5 for $q\Delta t^4/r = 10^{-4}$, 10^{-2}, 1, 10^2 and 10^4 respectively. As in Chapter 4 the initial state vector was known perfectly, and the filters were run until the track loss criteria of section 4.4 were satisfied. N_{AVE} is the average of 100 replications and 95% confidence limits are shown with each point (assuming track lifetime is exponentially distributed). Also shown is the average track lifetime N_L for a constant velocity prediction on the basis of the perfectly known initial state vector. For this prediction measurements are ignored, so that the average track lifetime N_L of the prediction estimate is independent of ρr . N_L should provide a lower limit on filter performance which may be approached as the relative density ρr of false measurements becomes large. Note that N_L increases as $q\Delta t^4/r$ decreases, *ie* as the normalized level of target manoeuvre decreases. The average number of mixture components before and after approximation is also shown in Figs 5.1 to 5.5. The average cpu time required to perform a single time step for the CAF and the PDAF is recorded in Table 5.1. For each pair of problem parameters, this table also indicates whether all replications were halted by just one of the two track loss criteria.

The parameter ρr is the density of false measurements relative to the true measurement error variance. However the difficulty of the tracking problem is likely to depend on the density of false measurements from the 'point of view' of the filter. Consider a single feasible track corresponding to a mixture component of the state pdf.

For given ρr, the number of false measurements that are plausible candidates for updating this track increases with $q\Delta t^4/r$, *ie* as the variance of target manoeuverability relative to r increases. On this basis a more appropriate measure of problem difficulty may be the average number of false measurements passed by a filter acceptance test, (see section 3.2). It is convenient to use the acceptance region based on the equivalent steady state Kalman filter problem, as this is independent of the values of individual measurements. The area A_∞ of this acceptance region is given by:

$$A_\infty = \frac{\pi r\, T_A}{1 - \alpha}$$

where α is the steady state value of the position Kalman gain and $T_A = 13.82$ is the acceptance threshold corresponding to a 99.9% chance of accepting the true measurement. It can be shown (see Bridgewater[49] for example) that α is given by:

$$\alpha = \frac{a}{2}\left(\frac{\sqrt{\phi}}{2} + 2 - a\right)$$

where $a = \sqrt{\frac{\phi}{4} + 2\sqrt{\phi}}$

and $\phi = \frac{q\Delta t^4}{r}$.

Thus the average number n_∞ of false measurements passed by this acceptance test is:

$$n_\infty = \frac{\pi \rho r T_A}{1 - \alpha} ,$$

which for given T_A depends only on ρr and $q\Delta t^4/r$. In Figs 5.1 to 5.5, the corresponding value of n_∞ is given with ρr for each of the results shown.

5.2.2 Discussion of results

The filters show similar performance trends in each of Figs 5.1 to 5.5. As would be expected, for given $q\Delta t^4/r$, track survival time increases as ρr and n_∞ decrease. Also the track survival time of the CAF approaches that of the PDAF for both small and large values of ρr. (This convergence for small ρr is not shown in Figs 5.1 and 5.2 as track survival time is so long in these cases, that the computation time for the simulation would be prohibitive.) Between these extremes, the CAF outperforms the PDAF. The average track lifetime of the CAF exceeds that of the PDAF by a factor of 10 in some cases, although an improvement factor between 3 and 5 is more common. The region of the ρr , $q\Delta t^4/r$ space where the CAF gives a significant improvement over the PDAF is sketched in Fig 5.6a. Although this diagram is only approximate, the region clearly depends on $q\Delta t^4/r$. In Fig 5.6b the region of improvement is sketched for the parameter space n_∞ , $q\Delta t^4/r$. In this space the dependency with $q\Delta t^4/r$ is not so strong, but is still quite evident. So performance of the CAF with respect to the PDAF is not solely determined by n_∞ .

As the filters' performance deteriorates for increasing ρr , so the average number of mixture components before approximation increases. This is the response of the filters to the increasing difficulty of the tracking problem. Eventually the relative density ρr of false

measurements becomes so great that the received measurements are of very little use to either filter; N_{AVE} approaches N_L, the average track lifetime for a simple prediction. In these circumstances the filters generate a large number of mixture components (often averaging over 100 before approximation) and consequently the average computation time for a single filter time step becomes very large, particularly for the CAF (see Table 5.1). It is quite possible that in these cases, performance of the CAF is being limited by N_T (compare with section 4.5.1). Table 5.1 also shows that for large ρr every track is lost to the excessive error check, Criterion (ii) (see section 4.4).

As ρr is reduced, the average number of mixture components retained by the Clustering Algorithm decreases towards the lower limit of a single Gaussian. Thus the CAF approximation approaches that of the PDAF, which explains the convergence of N_{AVE} for the two filters. Note, however, that in several cases where the average number of mixture components after reduction for the CAF is only fractionally above unity, the average track lifetime for the CAF is about three times that of the PDAF. Also, in these cases, Table 5.1 shows that the average CAF computation time per filter iteration is only about twice that of the PDAF. The convergence of N_{AVE} for the CAF and PDAF with decreasing ρr can be clearly seen in Figs 5.3 to 5.5. The same effect may be expected for $q\Delta t^4/r = 10^{-2}$ and 10^{-4}, but as already explained the computation time for the necessary simulations is prohibitive. As ρr decreases, the average number of components before approximation tends to 2 for both filters. One of these components corresponds to an accepted measurement (nearly always the true measurement), while the other corresponds to the prediction which allows for the possibility that the true measurement has been rejected.

5.3 An example of filter operation

In this section, to gain an insight into the operation and performance of the CAF and the PDAF, a single run of the tracking filters is examined in detail. The chosen example has the following parameters:

$$P_D = 1$$

$$\frac{q\Delta t^4}{r} = 1$$

and

$$\rho r = 0.005 \;,$$

which gives $n_\infty = 0.8683$. Also the maximum number of components allowed after approximation by the Clustering Algorithm was set at $N_T = 10$. These parameters determine filter performance. To generate an interesting target trajectory the initial speed was chosen to be $u_0 = 10\sqrt{r}/\Delta t$, initial target heading was chosen randomly and the initial position was the origin. Fifty time steps of tracking have been simulated.

5.3.1 Filter tracking performance

For this example, the target position at each time step is shown in Fig 5.7, together with the tracks or position estimates (*ie* the mean of the pdf of target position) of the CAF and the PDAF. The true measurement generated at each time step is also shown, although the false measurements have not been plotted. The units of the X and Y axes are normalized with respect to $u_0\Delta t$, and the scale of the Y axis is slightly stretched.

The first few estimates of the filters are very accurate since the initial target state vector is given. The CAF position estimate follows the target quite well throughout and the most noticeable errors occur at target manoeuvres. The tracks of the two filters are very similar up to about time step 17, at which point the PDAF estimate diverges from the trajectory. The PDAF apparently regains track (probably fortuitously) at time step 24, but fails to follow the subsequent sharp target maneouvre and soon finally diverges from the target trajectory. The point at which the PDAF track fulfils the second track loss criterion of section 4.4 is shown on the diagram. As expected from the track plot, the CAF estimate does not alert either of the track loss criteria.

To provide a precise record of the tracking error history, plots of the estimation error in position and velocity are shown in Figs 5.8 and 5.9 for the CAF and PDAF respectively. The magnitude of the actual position error at time step k is calculated from:

$$\sqrt{\left(\hat{x}_k - x_k\right)^2 + \left(\hat{y}_k - y_k\right)^2}$$

where $(\hat{x}_k, \hat{y}_k)$ is the estimate of the target position at time step k and (x_k, y_k) is the actual target position at time step k. The calculation of the velocity error is similar. In addition to the actual error, an indication of the filter's own view of its estimation error is shown as a dashed line. The measure of error (denoted the predicted error in Figs 5.8 and 5.9) is derived from the overall covariance matrix of target state, and at time step k it is given by:

$$\sqrt{P_{x_k} + P_{y_k}}$$

where P_{x_k} and P_{y_k} are the diagonal elements of the covariance matrix corresponding to target position. The measure of the velocity error is similar. Also shown for reference is the error measure (obtained from the covariance matrix, as above) for the equivalent Kalman filter, *ie* for the optimal filter in the absence of false measurements. Note that the *square* of the predicted error measure for the filters is the expected value of the actual error magnitude squared.

While the Kalman filter predicted error measure rapidly reaches a steady state, the predicted errors for the PDAF and the CAF vary throughout the track and are always greater than or equal to the Kalman filter reference. This is because the covariances for the PDAF and the CAF, which must operate with uncertain measurement association, depend upon the values of the received measurements. However, the covariance of the Kalman filter, which assumes that only true measurements are received, is independent of any measurement values. The predicted error measures of the PDAF and the CAF cannot be better than that of the Kalman filter since the latter is not corrupted by false measurements (see Ref 11).

The actual estimation errors of the CAF (Fig 5.8) show large fluctuations, but there is no trend of increasing error through the track. There are clear peaks in the position and velocity tracking errors at time step 25, when the target executed a sharp turn. At each of these maxima, the CAF's predicted error measure also peaks and closely matches the actual error. Throughout the track, the CAF predicted error is of the same order as the actual error and on several occassions significant peaks coincide or are very close. Clearly, through

statistical fluctuation, a perfect match over the whole track is not expected.

For the PDAF, a sharp rise in position error following track loss is clearly shown in Fig 5.9. The filter's predicted position error also rises, but is much smaller than the actual error by the end of the track.

5.3.2 Filter operation

The CAF estimate of target state at a given time step is the mean of a Gaussian mixture distribution, each component of which corresponds to a feasible target track. As explained in Chapter 2, if several measurements passed by the course acceptance test, these tracks subdivide so producing a tree like pattern of potential tracks which are controlled by the Clustering Algorithm. The growth of potential tracks for the current example is illustrated in Fig 5.10. The overall CAF estimate is shown as a dashed line, the PDAF estimate is shown as a continuous black line and the actual course of the target is shown by small circles. The potential tracks, after the clustering operation, are shown as coloured lines, the colour of the line indicating the weighting of the track (*ie* the probability that this is the correct track). To show the potential tracks in the vicinity of the target loop (labelled f on the track in Fig 5.10) more clearly, this part of the picture has been enlarged and slightly stretched in Fig 5.11, approximately by a factor of 6.

The number of potential tracks varies considerably over the history of the track. It appears that the number of tracks increases

when the target executes a manoeuvre (see points c and f on the target trajectory in Fig 5.10). This is because the target model gives the expected advance of the target as a straight line, and so tentative tracks into false measurements are produced. These extra tracks are eliminated when a steady course has been resumed (points b, d and g in Fig 5.10), showing that the Clustering Algorithm is economical in its management of potential tracks.

Throughout most of the track history, at least one of the potential target tracks closely follows the path of the target, and so has probably correctly selected the true measurements. Also note that when a tentative track with β weight above 0.5 is produced (green line), this track is almost always close to the actual target path. At times when the filter appears to have difficulty in maintaining track, usually no potential track with a large β weight is produced (see Fig 5.11).

At point c and in the vicinity of point f on the target path (Fig 5.10), the PDAF estimate diverges from the actual trajectory. At these points, the Clustering Algorithm has allowed the growth of diverging potential tracks, each with a significant β weighting. Fig 5.12 shows contours of the approximated position pdfs of both the CAF and the PDAF at the 17th time step (the point after label c in Fig 5.10). The actual position of the target is also marked and it is clearly associated with the dominant cluster component, which accounts for 85% of the total probability mass of the mixture. The second most important cluster component has a β weight of 0.12. The PDAF single Gaussian approximation appears to be stretched between these two major components. The PDAF approximation is the result of a separate track propagation and approximation sequence, although up to this time, the PDAF and CAF tracks are similar and so it is likely that

the PDAF generates a fairly similar posterior pdf before approximation at this time step.

To show how the position pdfs evolve, contours for the following two time steps (18 and 19) are shown in Figs 5.13 and 5.14. At time step 18 (Fig 5.13), after clustering there are now only two components. The weak component of the previous time step has been eliminated. The single Gaussian of the PDAF has been further stretched and flattened so that its centre still lies between the two cluster components, but has moved further away from the dominant component. At time step 19 (Fig 5.14) only a single Cluster Algorithm component is retained, which is sharply concentrated on the target path. The PDAF approximation is now well removed from the true path but still retains the elongated form as a legacy of time step 17, but which is no longer relevant. This illustration shows the importance of retaining more than one component at critical times during the tracking operation.

The situation six time steps later (time step 25) is shown in Fig 5.15. This is close to the label f in Fig 5.10 and here the CAF is propagating two main clumps composed of eight components. The PDAF has recovered from its poor pdf approximation at time step 19 (possibly through a fortuitous absence of false measurements in the track vicinity) and again straddles the CAF mixture pdf. However, as can be seen in Fig 5.10, subsequently the PDAF fails to follow the target manoeuvre and the track is lost for good. The single Gaussian approximation cannot cope with two diverging branches, each with significant weighting.

Figs 5.16 and 5.17 show how the number of components of the mixture distribution varies, and also the values of the most significant 8 weights at each time step for the CAF and the PDAF. For the CAF, the

values of the five largest β weightings *after* clustering are shown as five time traces; whereas for the PDAF, the five largest β weightings *before* approximation are shown. At each time step the β weights have been ordered in decreasing magnitude, so the BETA1 trace always shows the largest value. Together with these traces, the number of mixture components before and after approximation has also been plotted.

Throughout most of the track, the Clustering Algorithm (Fig 5.16) keeps the number of components after approximation well below the allowed limit of $N_T = 10$. Comparing Fig 5.8 with Fig 5.16, it can be seen that when the filter is tracking well, the number of components is kept low. Only when tracking becomes difficult, such as during the target loop in this example, does the number of components rise and significant β weighting extend to more than three components after clustering. In Fig 5.17 it can be seen that as the PDAF became lost, the number of components before approximation rose greatly and eventually reached a maximum of 1070. This increase is due to the expansion of the filter's acceptance region, which accompanies an increase in the tracking error as perceived by the filter. Before track loss, the β weighting traces show that there are usually only two or three significant components, one of which is usually clearly dominant. As the PDAF begins to lose track, the dominance of any one component declines, and the BETA4 and BETA5 traces show a temporary increase. As the number of components rises well above five, all the BETA1 to BETA5 traces fall towards zero as the weighting is shared amongest many components. This indicates that at each time step the filter has generated many hypotheses, each of which has a very small probability of being correct.

Finally, as an illustration of how the CAF responds to losing track, the potential tracks produced by a different example are shown in Fig 5.18. The parameters of this example are the same as the previous case, except the density of false measurements has been doubled. When the target manoeuvres, the filter's tracks split into two diverging branches, one of which continues on the original target heading while the other follows the target manoeuvre. However this latter branch eventually dies out. This is probably due to the true measurements having a similar, unusually large error on several consecutive time steps, while by chance false measurements fell close to the predicted target positions on the other branch. Note that after loss of track there is a tendency to produce diverging tracks with small β weights, and tracks with β weights above 0.5 are only produced on two time steps out of twenty-three.

5.4 Discussion and conclusions

In section 5.2, the performance of the CAF has been compared with that of the PDAF for the standard example of the baseline problem (second order target model with true and false Cartesian position measurements). The results presented in Figs 5.1 to 5.6 should enable one to obtain an initial assessment of filter performance for a variety of two-dimensional tracking problems. Even if the required problem is not of exactly the same form as the standard baseline case, it may be possible to derive a rough correspondence so that approximate values for the equivalent baseline problem parameters may be found. An indication of the average track lifetimes for the CAF and the PDAF with the required parameter values may be obtained by interpolation or extrapolation from the presented results. This should show whether the performance of the PDAF is likely to be adequate for the application, and if not, whether the CAF can provide

the necessary improvement. Clearly a detailed simulation should be carried out to confirm this initial assessment before any implentation is attempted.

The values of the average track lifetime given in section 5.2 depend on the definition of track loss (see section 4.4). A track is counted as lost if, over five consecutive time steps, either

(i) the true measurement is rejected,

or (ii) the tracking error is 'large'.

These criteria may not be appropriate for all applications. For instance in Ref 11 track loss is only based on consistent rejection of the true measurement, and it is independent of tracking error (criterion (ii)). Under this reduced definition of track loss, the average track lifetime would be much greater than that shown in our results. This is especially so for the higher values of ρr, as in these cases track loss for all replications was due to criterion (ii) (see Table 5.1).

Table 5.1

PROCESSOR TIMINGS AND TRACK LOSS CRITERION

Problem parameters			Computer cpu time for one time step (ms)		Criterion on which tracks were lost	
$\frac{q\Delta t^4}{r}$	ρr	n_∞	CAF	PDAF	CAF	PDAF
10^{-4}	5×10^{-2}	2.50	4.32	0.19	i and ii	i and ii
	10^{-1}	5.00	7.77	0.22	ii only	ii only
	2×10^{-1}	10.00	17.14	0.35	ii only	ii only
	5×10^{-1}	26.01	64.11	1.23	ii only	ii only
	1	50.01	93.13	4.15	ii only	ii only
10^{-2}	10^{-2}	0.68	1.50	0.19	i and ii	i and ii
	2×10^{-2}	1.36	2.91	0.19	i and ii	i and ii
	5×10^{-2}	3.39	10.41	0.32	i and ii	i and ii
	10^{-1}	6.78	52.25	1.30	ii only	ii only
	2×10^{-1}	13.57	157.48	2.26	ii only	ii only
1	2×10^{-4}	0.035	0.32	0.15	i and ii	i and ii
	5×10^{-4}	0.087	0.33	0.15	i and ii	i and ii
	10^{-3}	0.174	0.34	0.16	i and ii	i and ii
	5×10^{-3}	0.868	1.08	0.24	i and ii	i and ii
	1.2×10^{-2}	2.084	5.93	1.12	i and ii	ii only
	2×10^{-2}	3.47	702.58	1.53	ii only	ii only
	4×10^{-2}	6.94	1267.19	3.27	ii only	ii only
10^{2}	10^{-5}	0.02	0.31	0.15	i and ii	i and ii
	10^{-4}	0.2	0.41	0.17	i and ii	i and ii
	5×10^{-4}	1.02	1.34	0.54	ii only	ii only
	10^{-3}	2.04	73.25	1.24	ii only	ii only
	2×10^{-3}	4.08	3012.38	2.77	ii only	ii only
10^{4}	10^{-8}	0.0012	0.30	0.15	i and ii	i and ii
	10^{-7}	0.012	0.30	0.15	i and ii	i and ii
	10^{-6}	0.12	0.30	0.16	i and ii	i and ii
	5×10^{-6}	0.59	0.88	0.27	i and ii	ii only
	10^{-5}	1.17	2.21	0.52	ii only	ii only
	2×10^{-5}	2.35	129.59	1.09	ii only	ii only

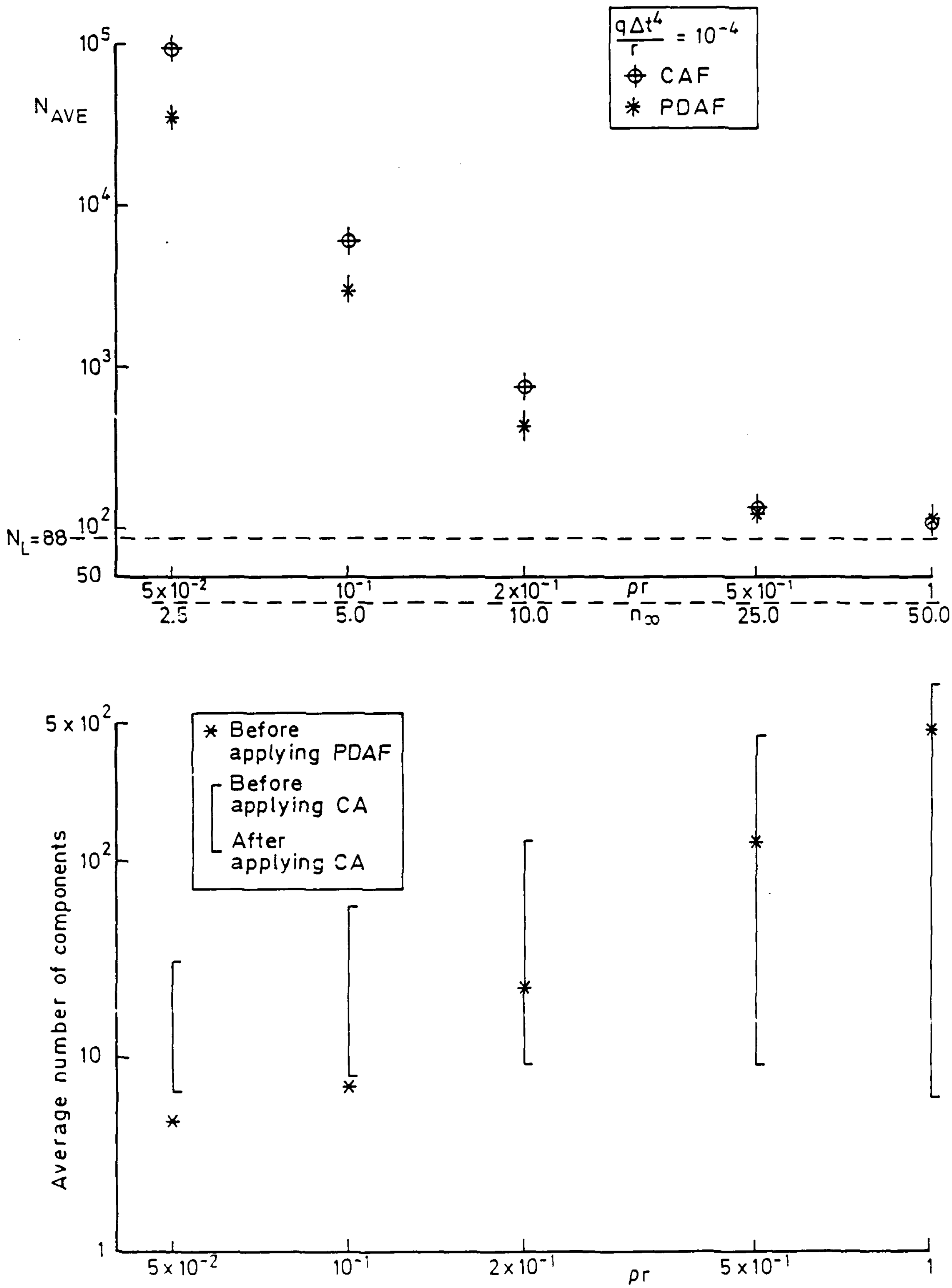


Fig 5.1 Performance of CAF and PDAF for $q\Delta t^4/r = 10^{-4}$

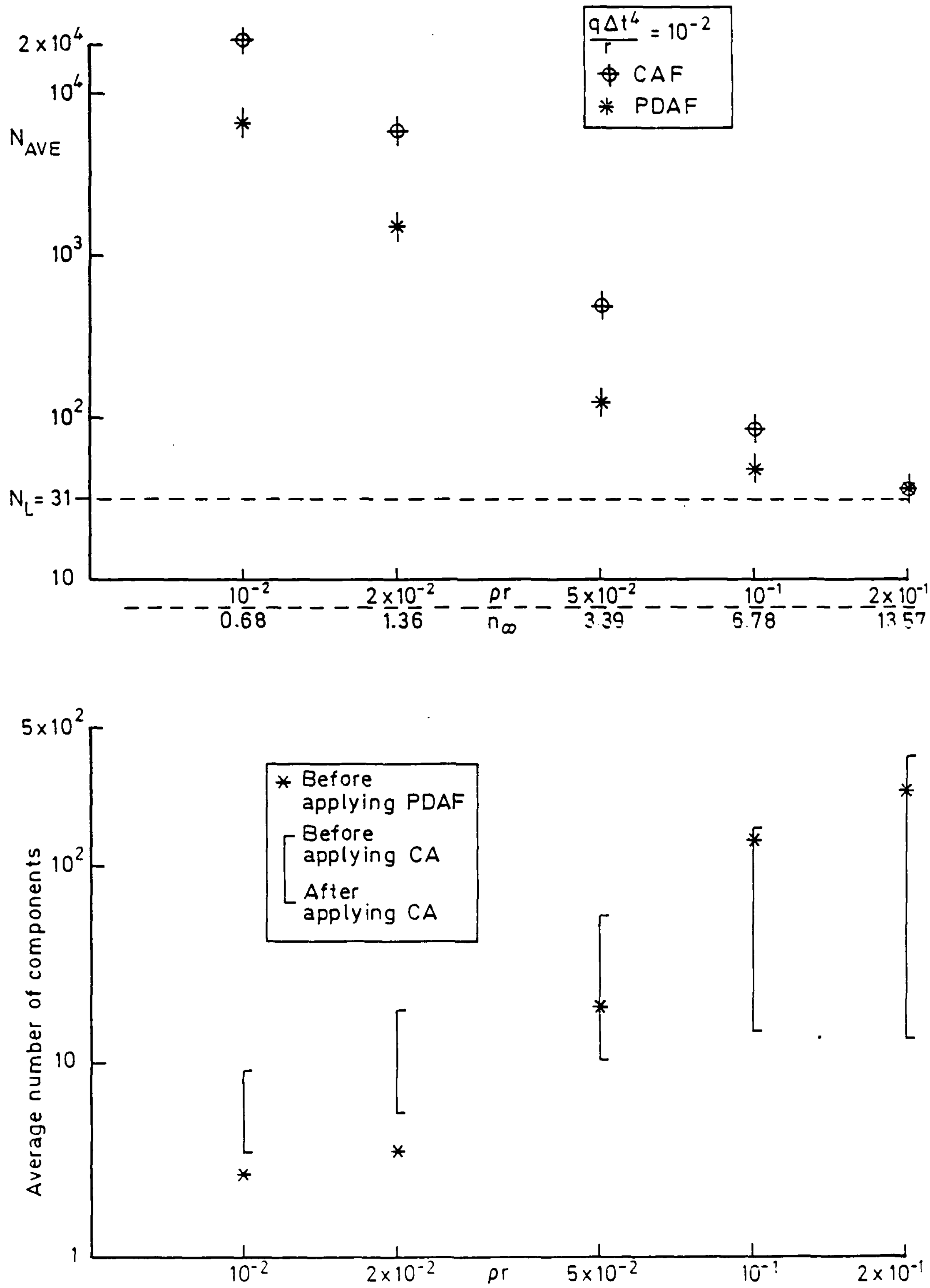


Fig 5.2 Performance of CAF and PDAF for $q\Delta t^4/r = 10^{-2}$

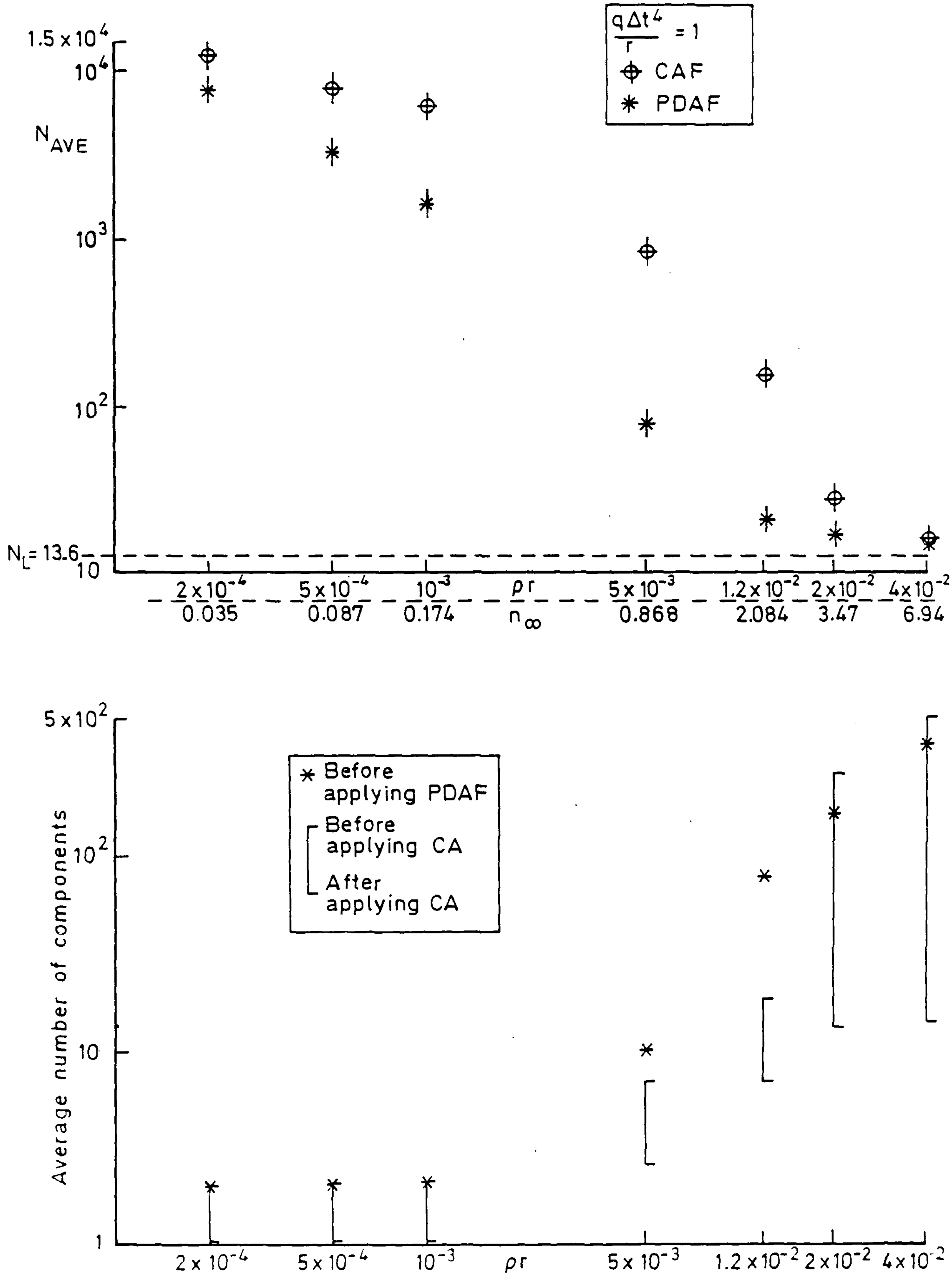


Fig 5.3 Performance of CAF and PDAF for $q\Delta t^4/r = 1$

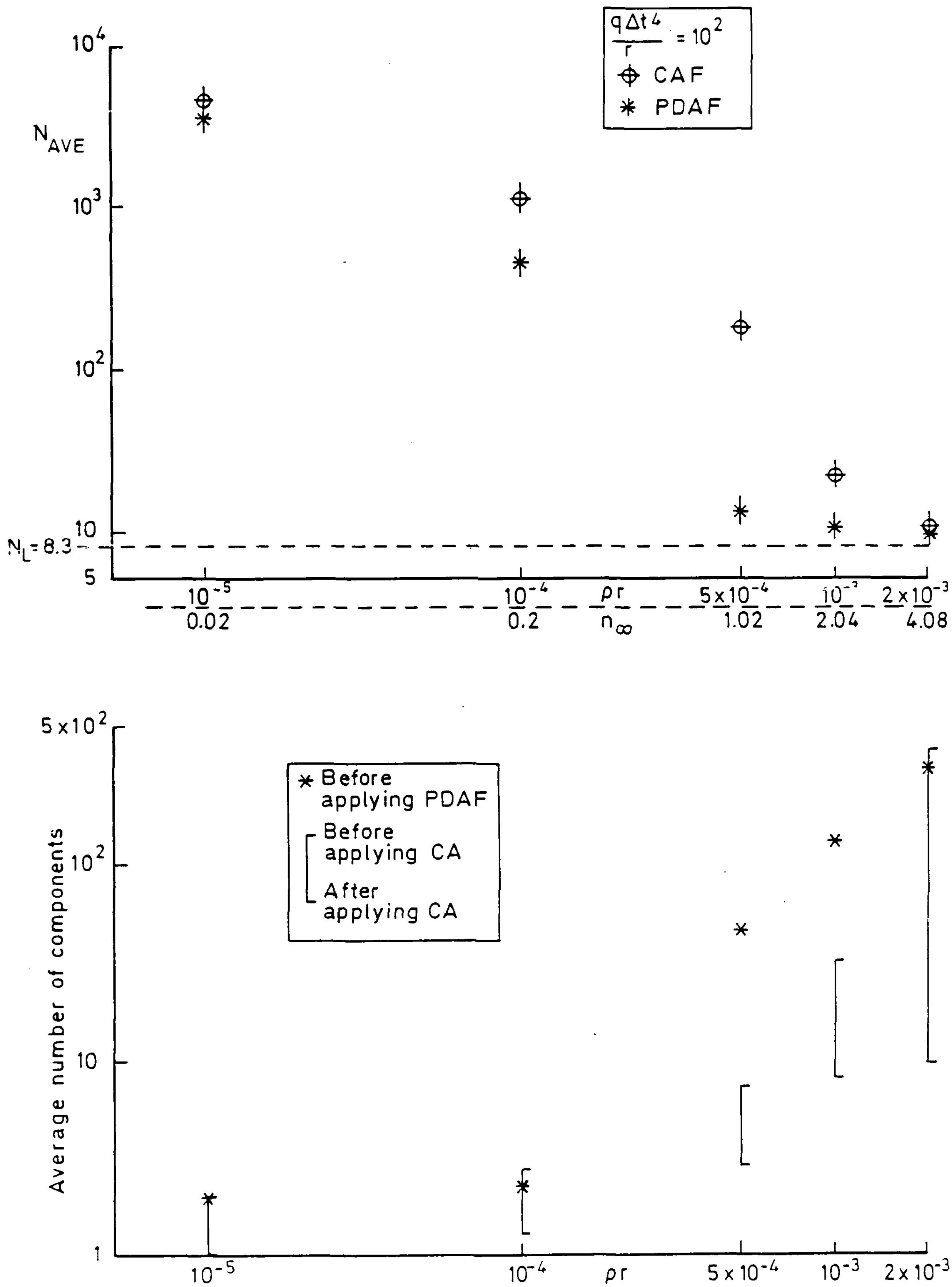


Fig 5.1 Performance of CAF and PDAF for $q\Delta t^4/r = 10^2$

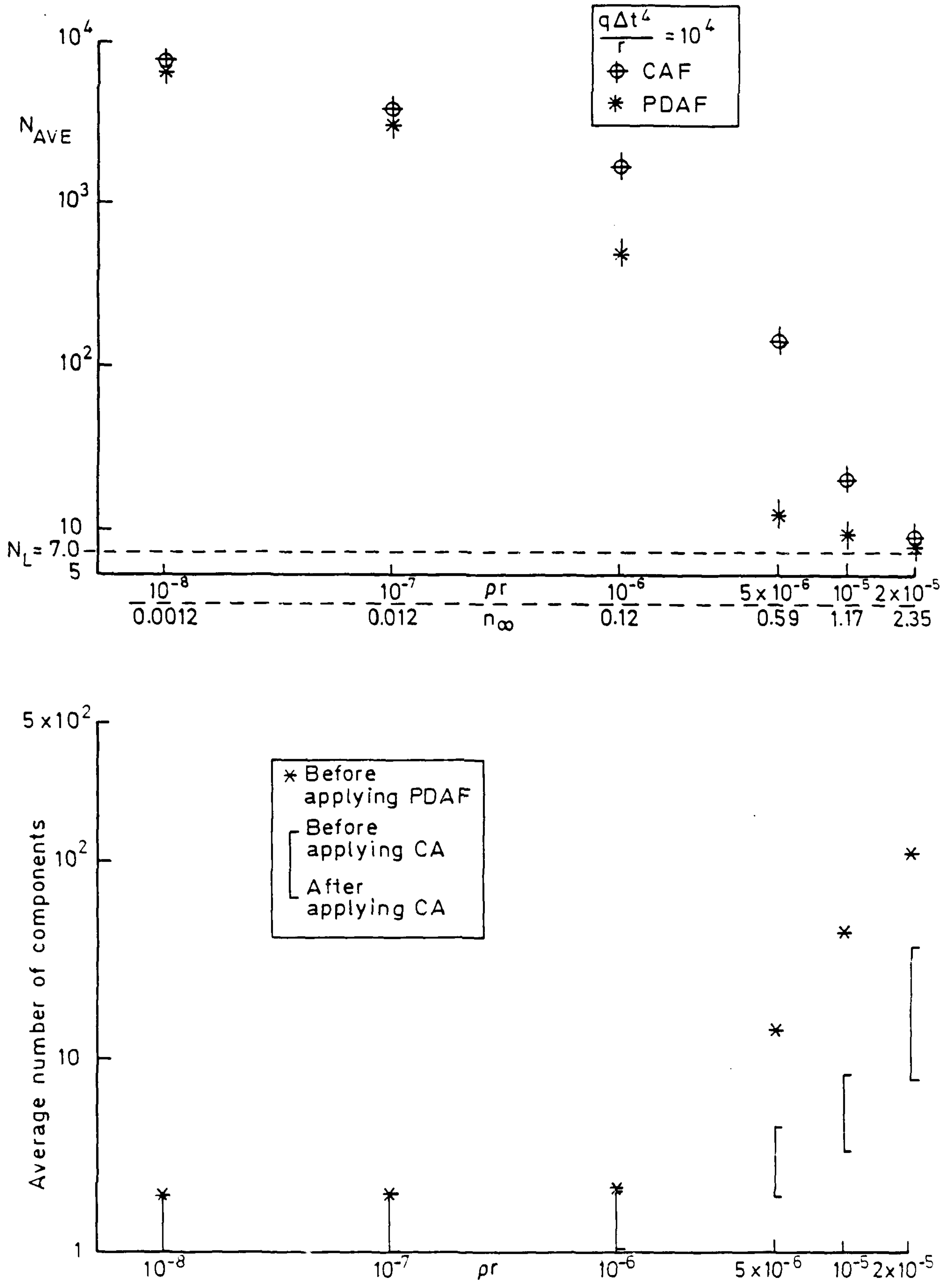


Fig 5.5 Performance of CAF and PDAF for $q\Delta t^4/r = 10^4$

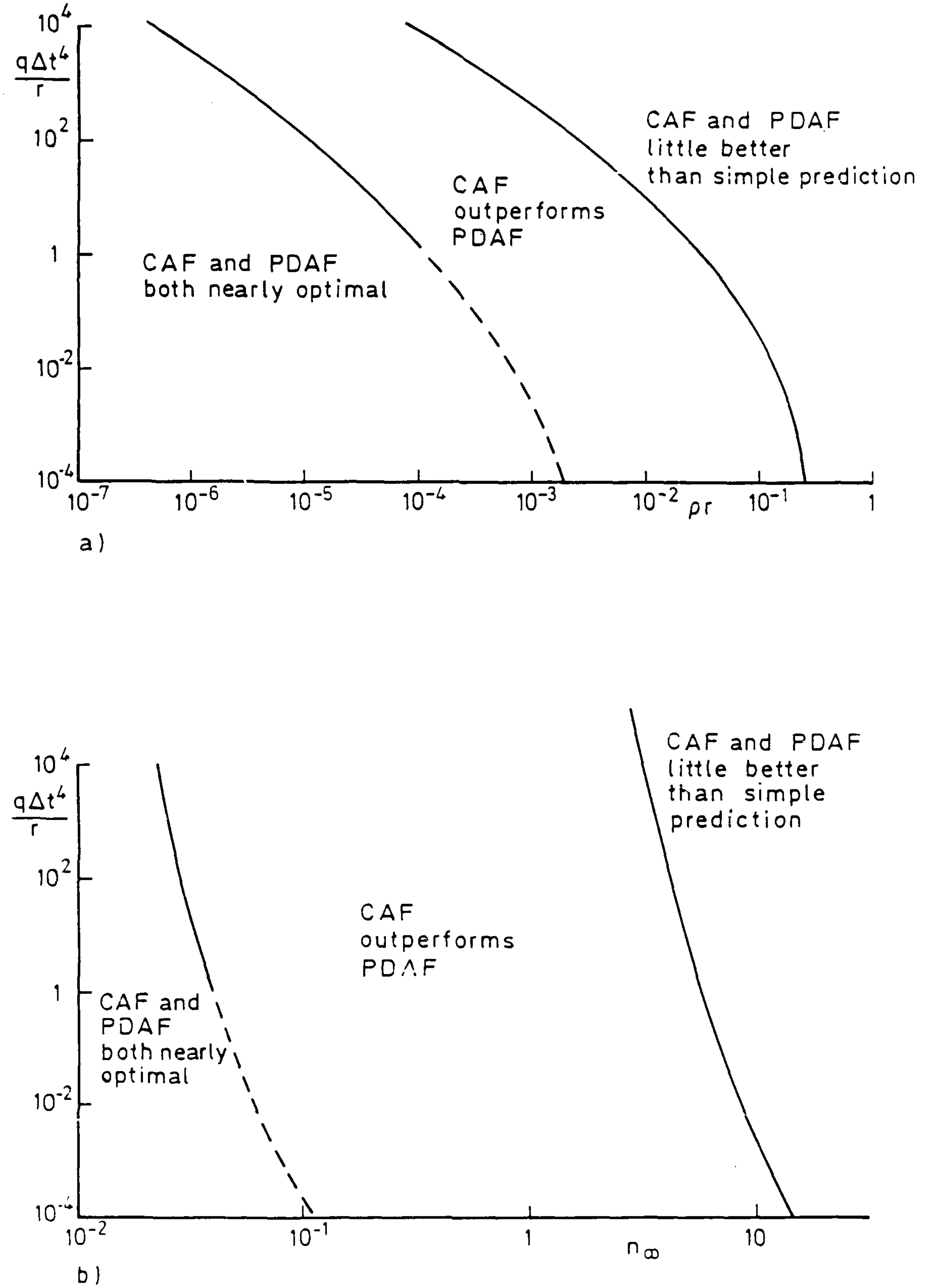


Fig 5.6 Filter performance over the problem parameter space

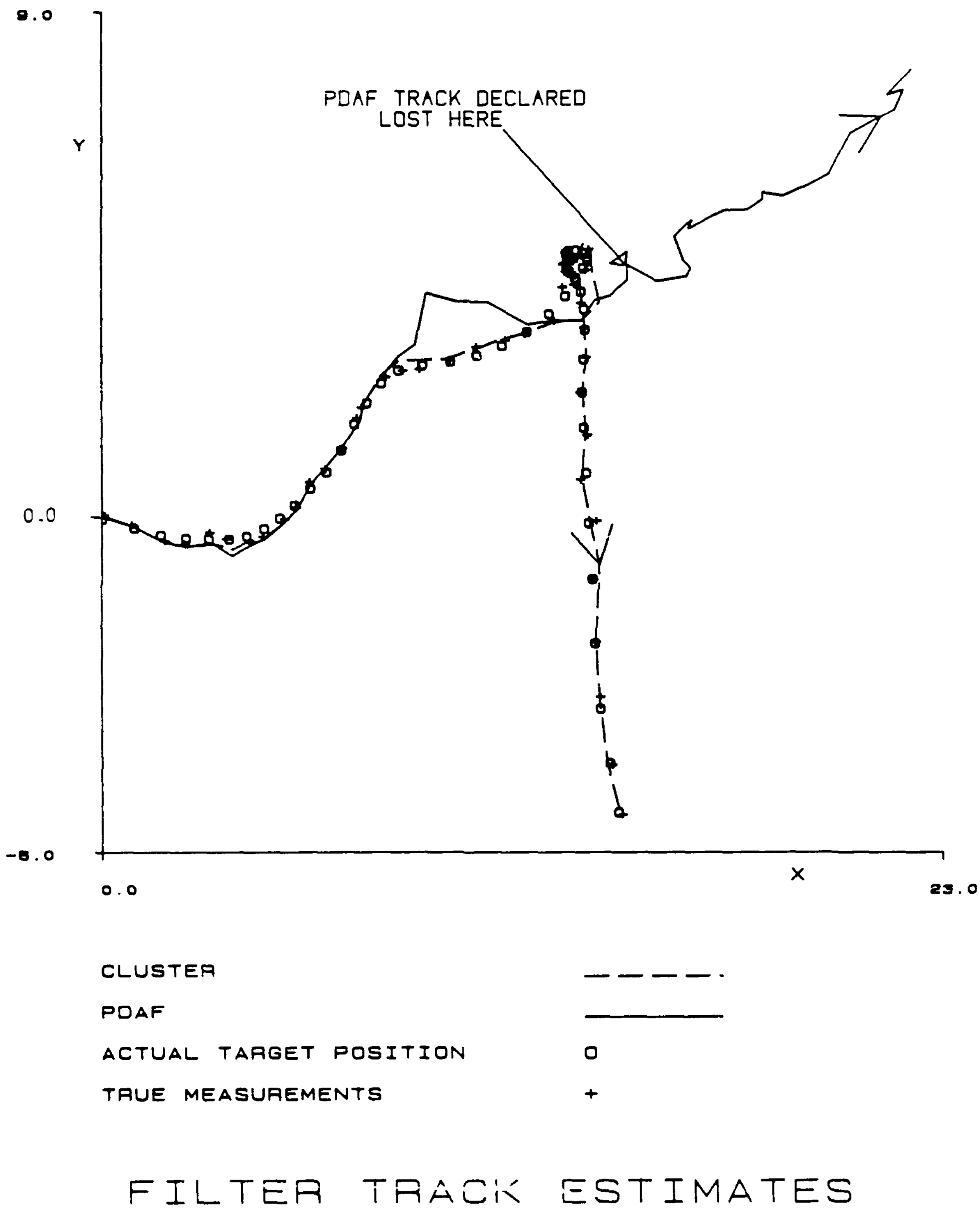


Fig 5.7 Tracking history of Clustering Algorithm filter and PDAF

CLUSTER ALGORITHM

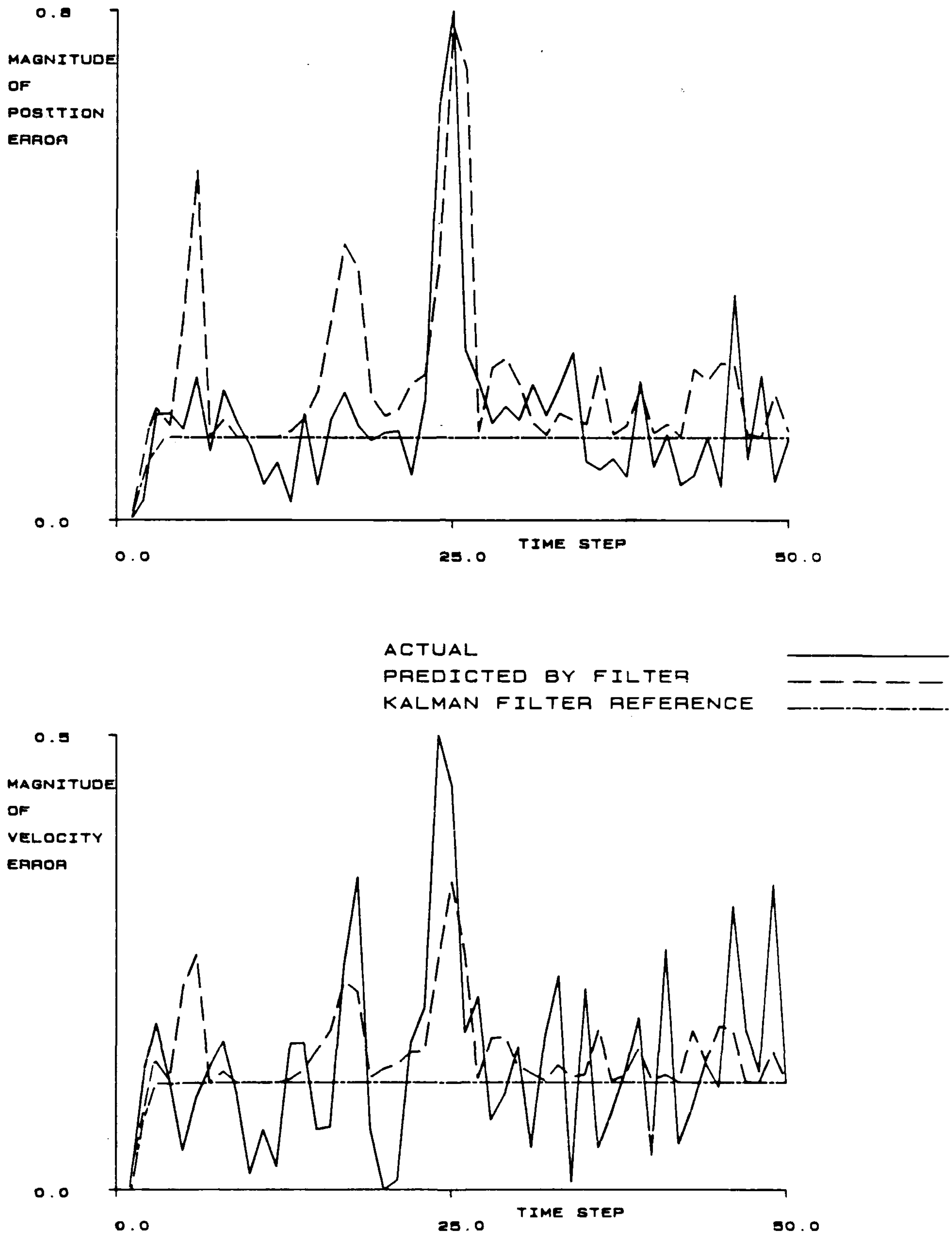


Fig 5.8 Tracking errors of Clustering Algorithm filter

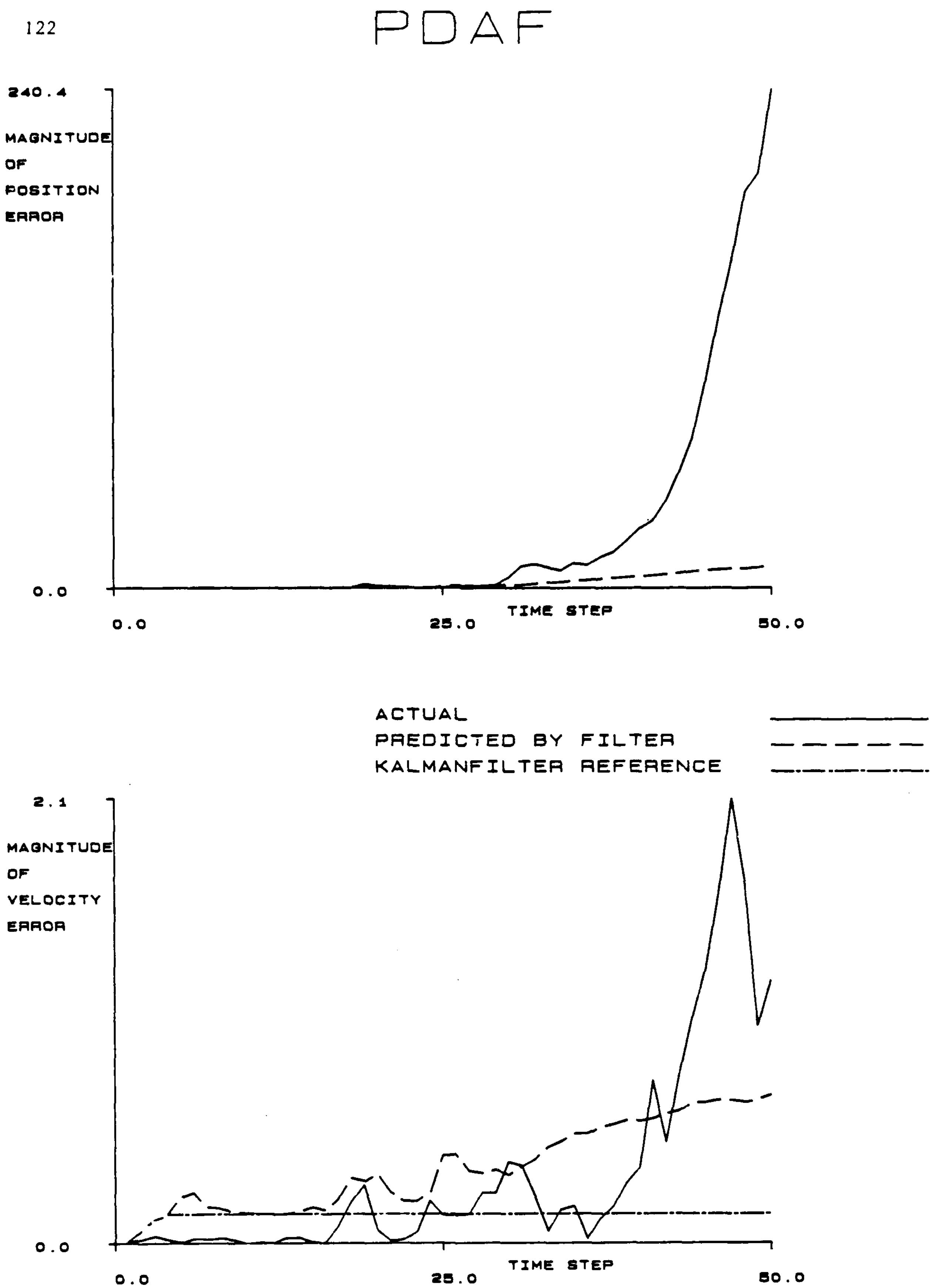
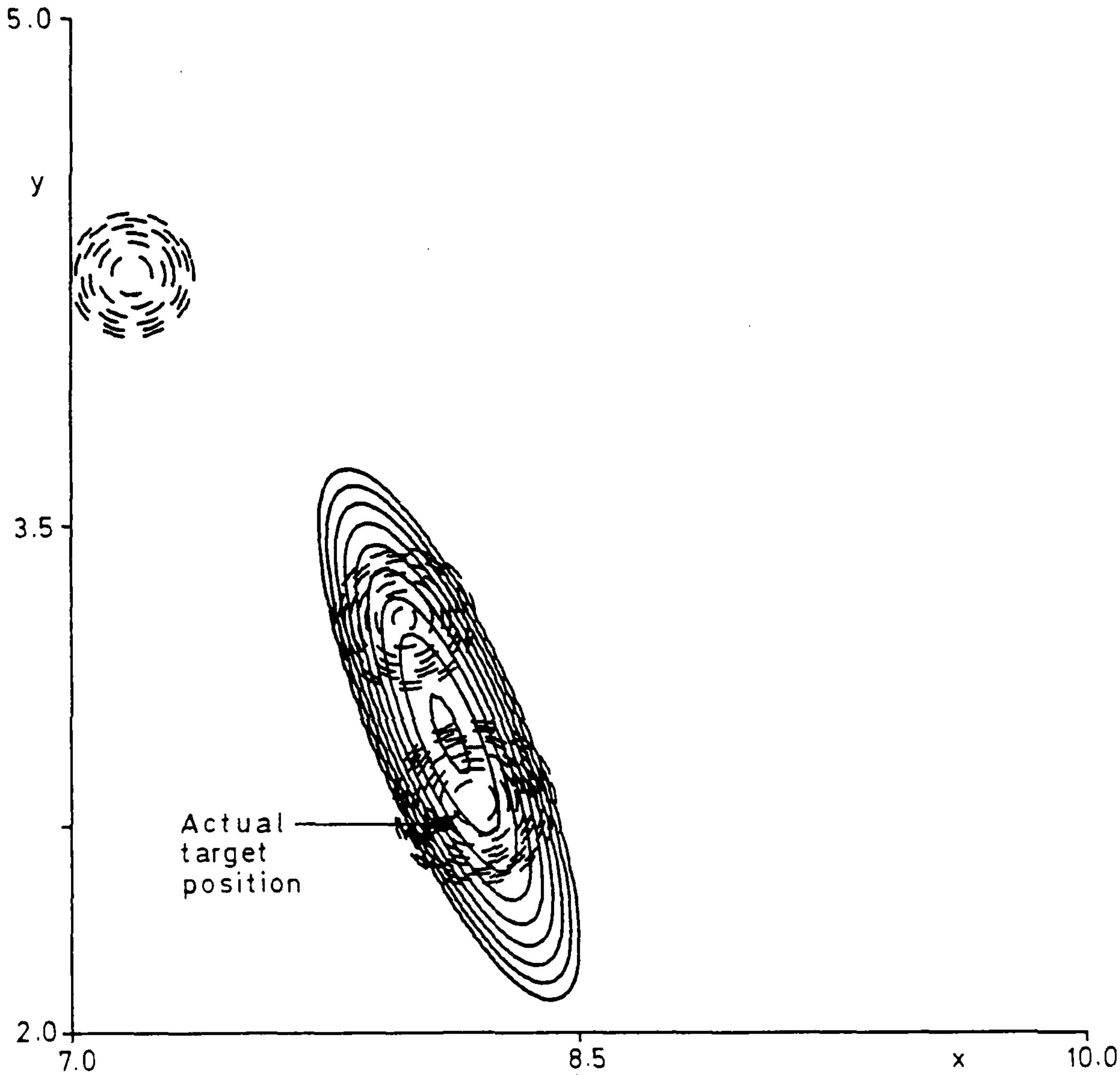


Fig 5.9 Tracking errors of PDAF

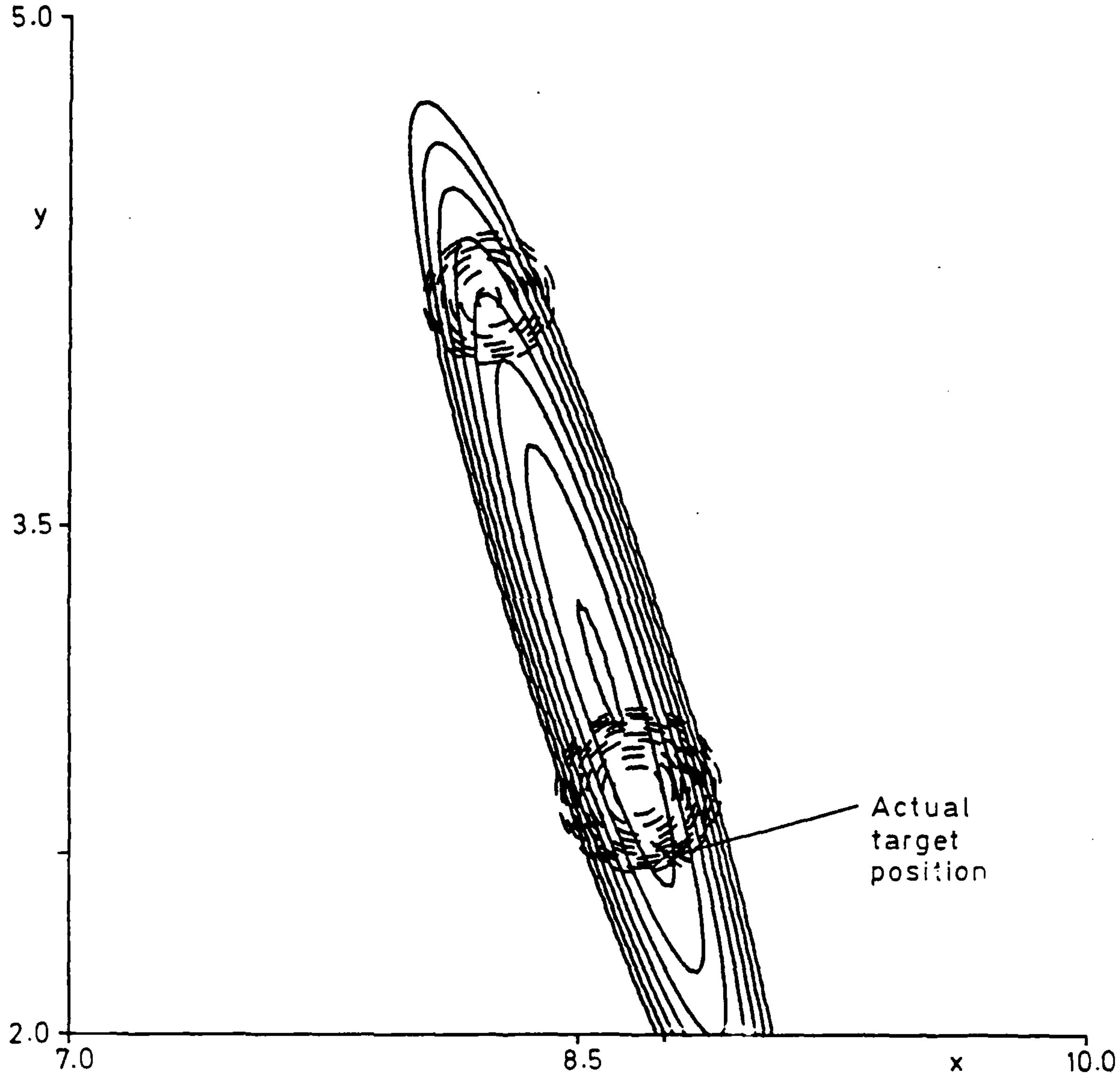


Logarithmic contour spacing
Contour heights are:

0.010000	0.021544	0.046416	0.100000	0.215443
0.464159	1.000000	2.154435	4.641589	10.000000

---------- Clustering Algorithm approximation
———— PDAF approximation

Fig 5.12 pdf of target position after approximation at the 17th time step

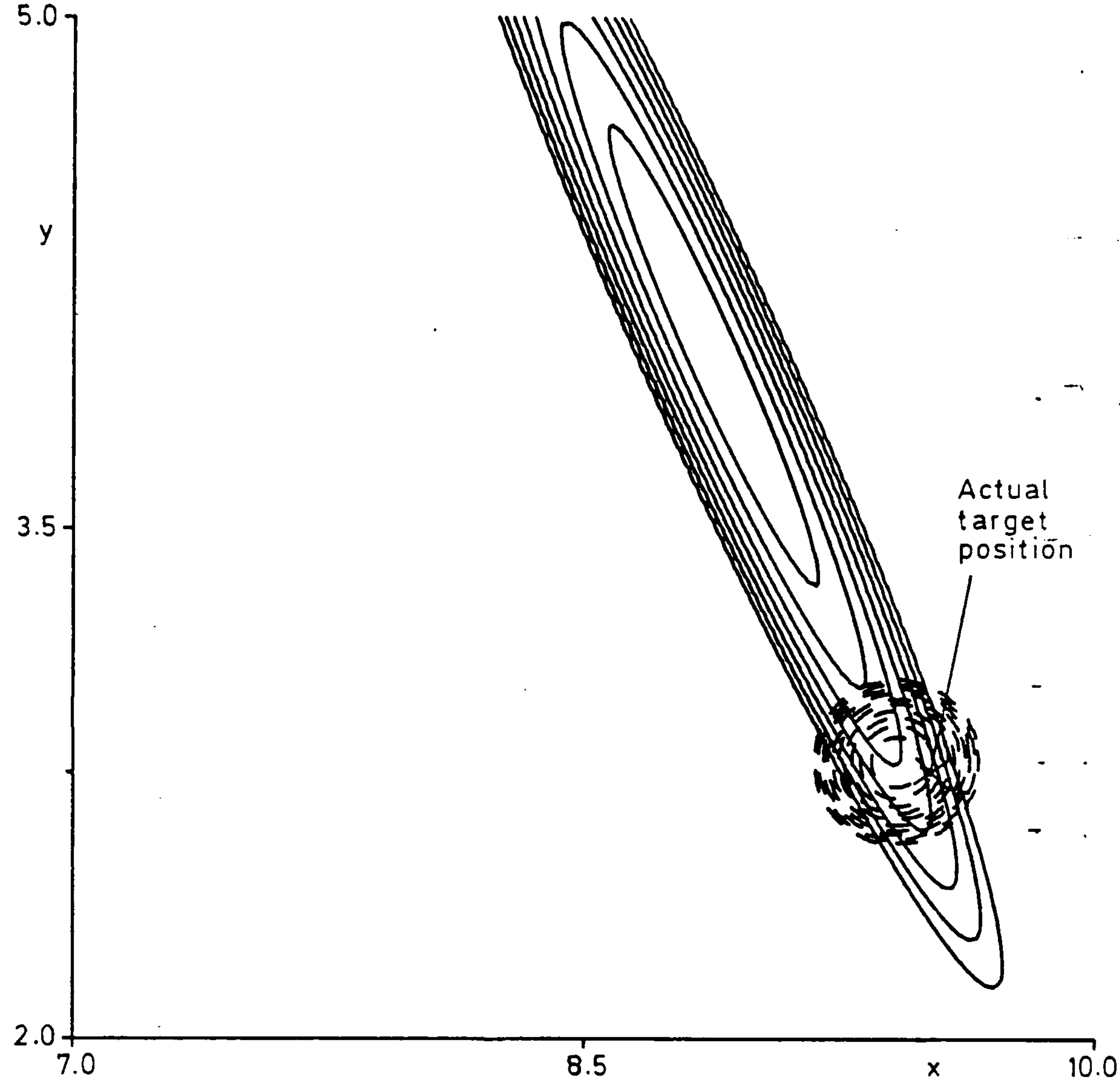


Logarithmic contour spacing
Contour heights are:

0.010000	0.021544	0.046416	0.100000	0.215443
0.464159	1.000000	2.154435	4.641589	10.000000

--------- Clustering Algorithm approximation
———— PDAF approximation

Fig 5.13 pdf of target position after approximation at the 18th time step



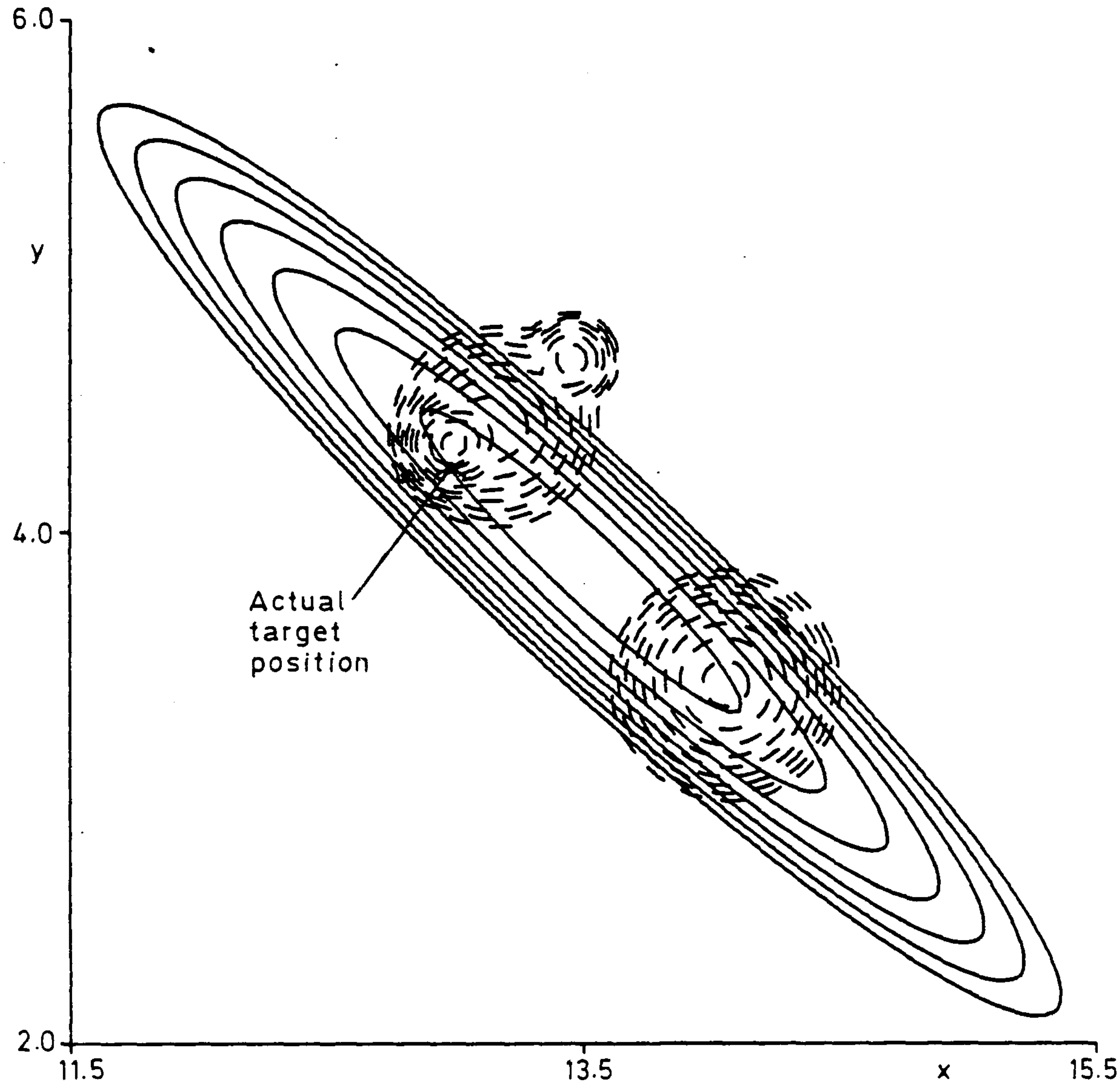
Logarithmic contour spacing
Contour heights are :

0.010000	0.021544	0.046416	0.100000	0.215443
0.464159	1.000000	2.154435	4.641589	10.000000

--------- Clustering Algorithm approximation

———— PDAF approximation

Fig 5.14 pdf of target position after approximation at the 19th time step



Logarithmic contour spacing
Contour heights are:

0.004642	0.010000	0.021544	0.046416	0.100000
0.215443	0.464159	1.000000	2.154435	4.641589

-------- Clustering Algorithm approximation
———— PDAF approximation

Fig 5.15 pdf of target position after approximation at the 25th time step

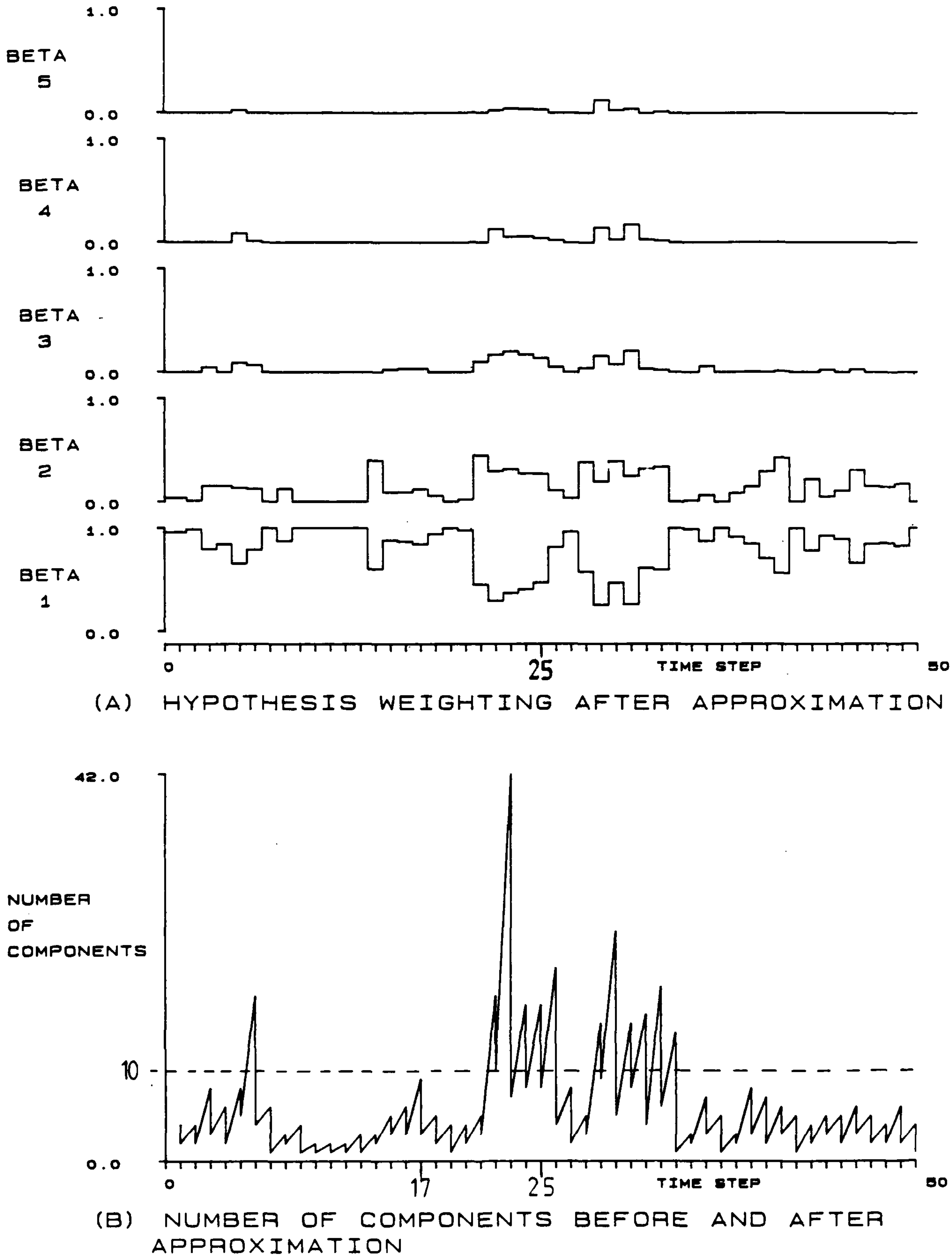


Fig 5.16 Number of components and β weightings for Clustering Algorithm filter

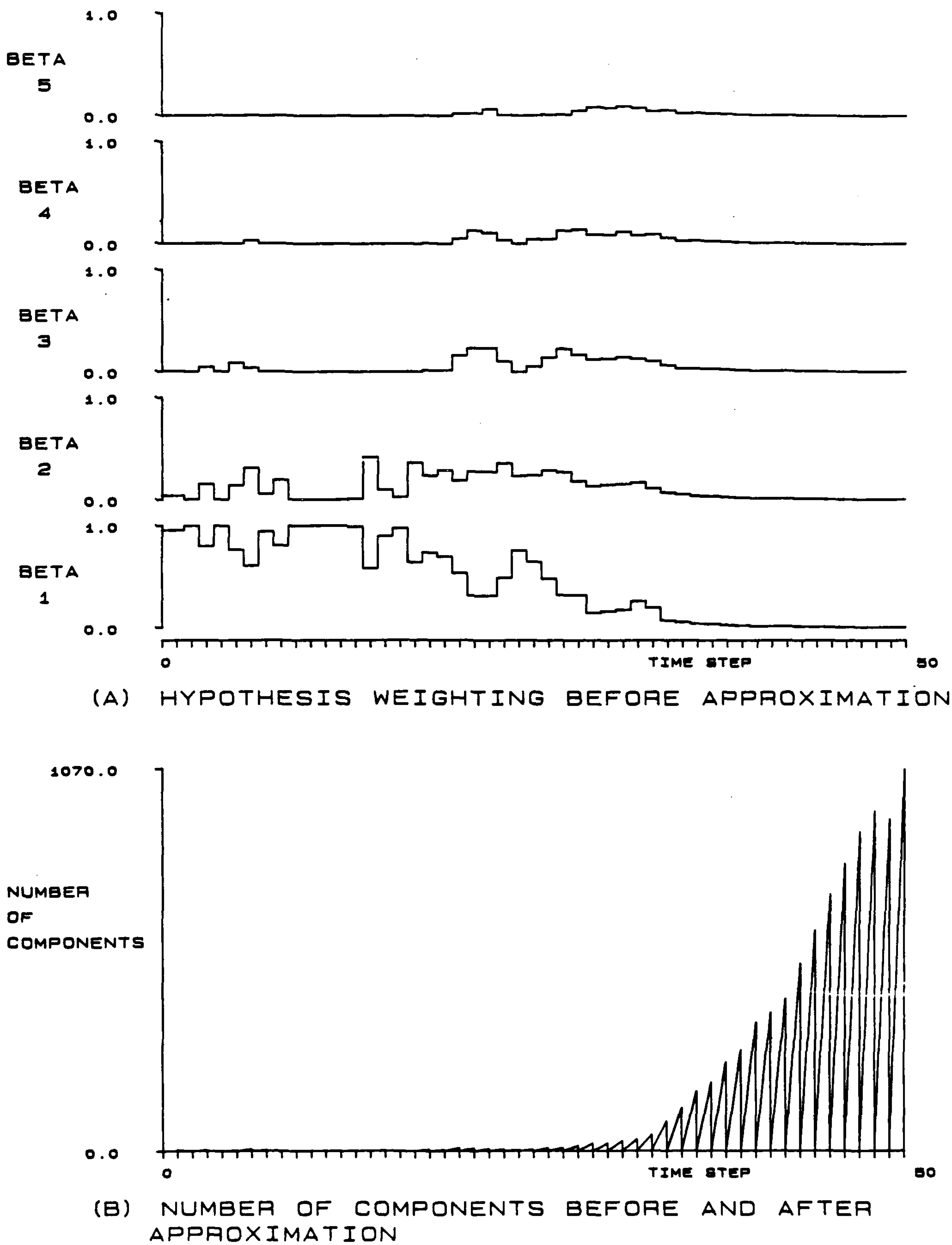


Fig 5.17 Number of components and β weightings for PDAF

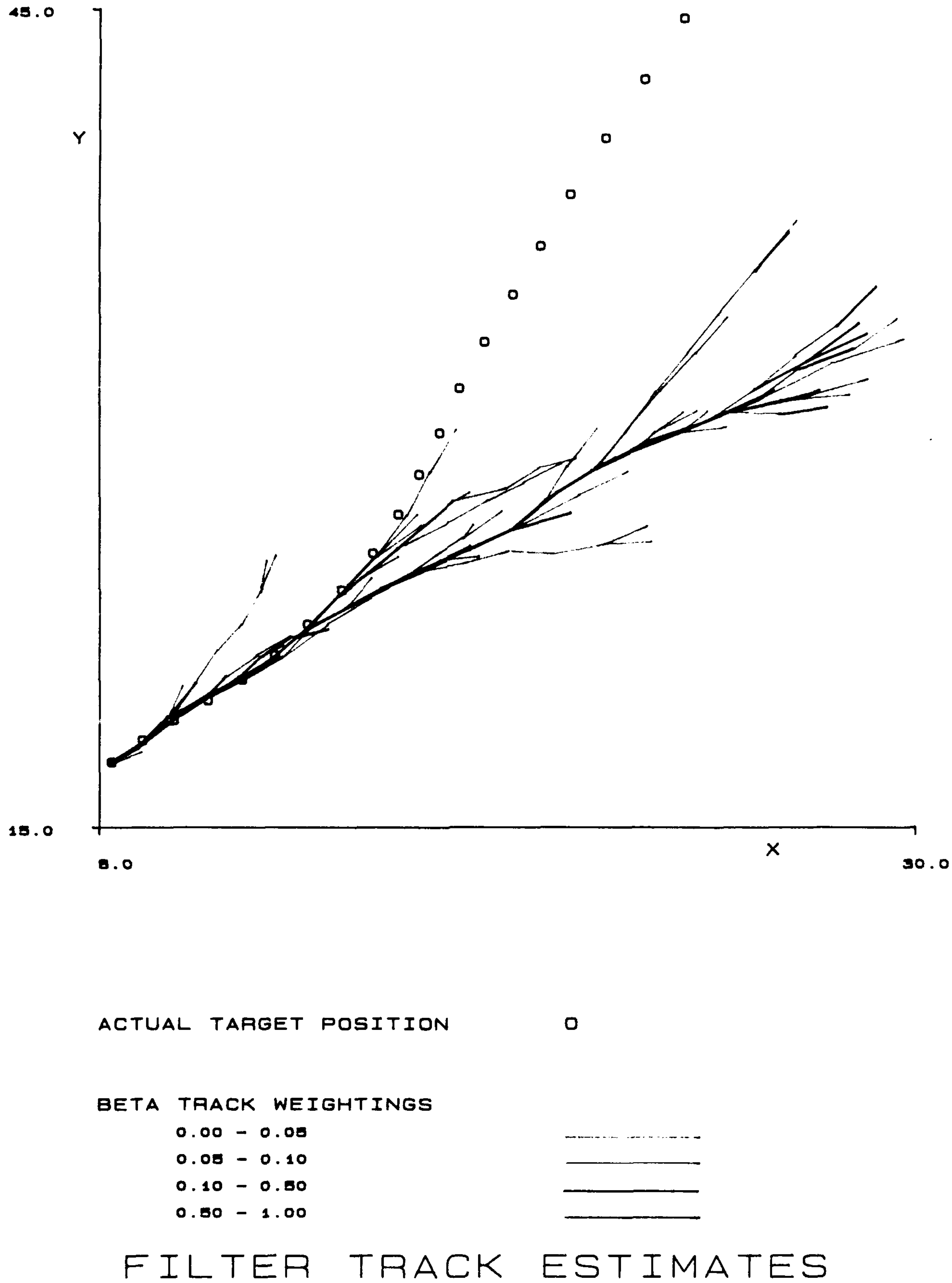


Fig 5.18 Feasible tracks produced by the Clustering Algorithm filter in the event of track loss

6 THE SECTOR SCAN PROBLEM

6.1 Introduction

The tracking problem of section 4.2 has been constructed so that performance depends on only a small number of non-dimensional parameters. This facilitates the assessment of filter performance over a wide variety tracking conditions (section 5.2). However this problem is somewhat unrealistic, principally because practical sensors, such as radars, usually produce measurements in polar co-ordinates rather than Cartesians. To show how this complication can be managed, a 'sector scan problem' has been devised. This example also serves to show how the assessment of section 5.2 can be used to give a rough indication of filter performance for a different tracking example.

The sector scan problem is to track a target passing through a surveillance sector in the presence of false measurements. A sensor at the origin produces position measurements in range and bearing, and false measurements are uniformly distributed in polar co-ordinates. On entering the sector, an initial estimate of the target position and velocity is supplied to the filter. (Note that the question of automatic track initiation is not considered in this study (see conference proceedings of Ref 49).) Since the target could enter the surveillance section from any direction, it is convenient to employ Cartesian state variables which allow the target kinematics to be represented by a linear model, in this case the usual second order model. This introduces a non-linear relationship between the state vector and the measurements, which complicates the filtering problem.

The sector scan problem is also used to investigate the effect on performance of target trajectories which are mismatched to the filter

model. This is another practical difficulty that must be considered in filter design. Two different types of mismatched trajectory have been examined:

(i) Trajectories simulated using the second order model with a value for the acceleration variance q which is different from that assumed by the filter.

(ii) Deterministic target paths consisting of periods of constant velocity motion and deliberate manoeuvres.

6.2 Problem description and solution

The surveillance sector is defined as the region where $X > 0$ km, $Y > 0$ km and:

$$2 \text{ km} < \sqrt{X^2 + Y^2} < 20 \text{ km} .$$

Every second, this region is scanned by a single sensor located at the origin, and a set of position measurements is passed to the tracking filter. These measurements are in polar co-ordinates. The probability of detecting a target that is within the sector is P_D, and the range and bearing errors on the true measurement are independent and Gaussian with zero mean. The variance of the range error is σ_r^2 and the variance of the bearing error is σ_θ^2. False measurements are uniformly distributed in polar co-ordinates. Thus the density of false measurements per unit area decreases with distance from the origin (see next section). Only one target is present within the sector.

When a target enters the surveillance region, the tracking filter is initialized with an estimate of target position and velocity. This

initial estimate has a Gaussian error of known covariance. The success of the filter in tracking the target is assessed by examining the position tracking error as the target is leaving the sector. Track is said to be maintained if:

$$\left.\begin{array}{l} |\hat{x} - x| \; < \; 10\,\sigma_x \\ \text{and} \\ |\hat{y} - y| \; < \; 10\,\sigma_y \end{array}\right\} \qquad (6.1)$$

where (x, y) are the co-ordinates of the target on the last sensor scan before the target leaves the sector, $(\hat{x}, \hat{y})$ is the corresponding filter estimate, and σ_x and σ_y are the standard deviations of the equivalent Kalman filter estimate (see later). This definition of track loss is derived from criterian (ii) of section 4.4. Clearly the tracking filter is not penalized for poor performance within the sector, but in practice it has been found that if the track deviates significantly from the target path, the filter is unlikely to regain track before the target leaves the sector.

The tracking filters which have been applied to this problem employ the usual second order target model (equation (4.1)) expressed in Cartesian co-ordinates, as this avoids the need for a non-linear model written in polar co-ordinates. However this does introduce a non-linearity between the true measurement and the target state vector. Thus equation (2.2) for the baseline problem statement should be replaced by:

$$\underline{z} = \underline{h}(\underline{x}) + \underline{v} \qquad (6.2)$$

where $\underline{z}$ is the true measurement, $\underline{x}$ is the state vector and $\underline{v}$ is the Gaussian measurement noise at some time step. For the present example:

$$\underline{z} = \begin{pmatrix} r_M \\ \\ \theta_M \end{pmatrix}$$

$$\underline{h}(\underline{x}) = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \\ \tan^{-1}(y/x) \end{pmatrix} \tag{6.3}$$

and the covariance of $\underline{v}$ is:

$$\begin{pmatrix} \sigma_r^2 & 0 \\ \\ 0 & \sigma_\theta^2 \end{pmatrix} .$$

The use of r to denote range should not cause any confusion with the measurement noise variance of the previous example.

If it is given that $\underline{z}$ is the true measurement and we attempt to apply the Bayesian techniques of Chapter 2 to this problem, the posterior pdf of $\underline{x}$ after updating with $\underline{z}$ will be non-Gaussian due to the non-linear element $\underline{h}(\underline{x})$. As the optimal Bayesian filter for this problem cannot be written in a simple recursive form, the sub-optimal extended Kalman filter (see Jazwinski[27]) has been employed. This filter is derived by linearizing about the state vector prediction

at each time step and then applying the standard Kalman filter relations. Thus at a given time step:

$$\underline{h}(\underline{x}) = \underline{h}(\bar{\underline{x}}) + \underline{\nabla}\ \underline{h}(\bar{\underline{x}})\ (\underline{x} - \bar{\underline{x}}) + \text{higher order terms} \quad , \tag{6.4}$$

where $\left[\underline{\nabla}\ \underline{h}(\bar{\underline{x}})\right] = \left(\left.\dfrac{\partial h_i}{\partial x_j}\right|_{\bar{\underline{x}}} \right)$ is the Jacobian matrix ,

h_i is the ith element of $\underline{h}(\underline{x})$

and x_j is the jth element of $\underline{x}$. For the present example, from equation (6.3):

$$\left[\underline{\nabla}\ \underline{h}(\bar{\underline{x}})\right] = \begin{pmatrix} \cos\bar{\theta} & 0 & \sin\bar{\theta} & 0 \\ -\dfrac{\sin\bar{\theta}}{\bar{r}} & 0 & \dfrac{\cos\bar{\theta}}{\bar{r}} & 0 \end{pmatrix} \tag{6.5}$$

where $\bar{r} = \sqrt{\bar{x}^2 + \bar{y}^2}$

and $\bar{\theta} = \tan^{-1}(\bar{y}/\bar{x})$. To derive the extended Kalman filter, the higher order terms in the Taylor expansion (6.4) are ignored. It can be shown that the resulting filter recursions are the same as those of the standard Kalman filter (equations (2.8)), but that the innovation vector $(\underline{z} - H\bar{\underline{x}})$ is replaced by $(\underline{z} - \underline{h}(\bar{\underline{x}}))$ and elsewhere H is replaced by $\left[\underline{\nabla}\ \underline{h}(\bar{\underline{x}})\right]$. In particular the covariance of the innovation is given by:

$$S = \left[\underline{\nabla}\ \underline{h}(\bar{\underline{x}})\right] M \left[\underline{\nabla}\ \underline{h}(\bar{\underline{x}})\right]^T + R \quad . \tag{6.6}$$

The output of the extended Kalman filter may be interpreted as the mean and covariance of a Gaussian approximation to the true posterior distribution. Thus when the false measurements are present, the extended Kalman filter may be used to propagate feasible tracks to make up a Gaussian mixture distribution for the target state. To evaluate the mixture weights of this distribution, the prior pdf of the true measurement for each track is required. This may be approximated by a Gaussian in polar co-ordinates with mean:

$$\underline{h}(\bar{\underline{x}}) = \begin{pmatrix} \bar{r} \\ \bar{\theta} \end{pmatrix}$$

and covariance S given by equation (6.6). Since the false measurements are uniformly distributed in polar co-ordinates, the mixture weights are given by equation (2.18) with $H\bar{\underline{x}}$ replaced by $\underline{h}(\bar{\underline{x}})$ and S given by equation (6.6). Clearly it is also convenient to carry out an acceptance test in polar co-ordinates, using this Gaussian as an approximation to the prior pdf of the true measurement. The filter may be implemented using the PDAF or the Clustering Algorithm approximation in the usual way.

6.3 Generation of target trajectories and measurements

Trajectories of targets passing through the surveillance sector may be generated either from the second order model as in the previous example, or deterministic trajectories consisting of constant velocity paths interspersed by deliberate manoeuvres may be generated. If the second order model is used, the variance q of the random numbers driving the model (the acceleration noise) may be chosen to be different from the model noise assumed by the filters. This allows the effect of parameter mismatch to be examined. The initial target heading on

entering the sector for each simulated trajectory is chosen at random from a uniform distribution over $[0, 2\pi]$, and the initial target speed is selected from a Gaussian distribution. The initial position is that point on the boundary of the sector for which the initial velocity vector passes through the centre of the sector. At each time step a true measurement may be simulated and false measurements of required density are generated over the complete sector. The simulation of a trajectory ends when the target passes out of the sector. Two separate random number sequences are employed. One of these is used for generating the target trajectory and the true measurements, while the other is used for generating false measurements. Thus the density of false measurements can be changed without altering the trajectories or the true measurements.

For this problem we shall not attempt to assess performance over a wide range of parameters, but the performance about a principal set of parameters will be investigated. For this principal problem, trajectories are generated using the second order model with $\Delta t = 1$ second and the standard deviation $\sqrt{q}$ of the driving acceleration noise chosen to be:

$$\sqrt{q} = 0.05 \text{ km sec}^{-2} \simeq 5\text{'g'} .$$

The initial target speed is drawn from a Gaussian with mean 0.3 km sec^{-1} and standard deviation 0.02 km sec^{-1}. Fig 6.1 shows a sample of eight trajectories generated with these parameters. For a sample of 100 trajectories, on average the target took 48 seconds to pass through the sector. True measurements produced by the sensor (a radar for example) have range errors of standard deviation $\sigma_r = 0.03$ km and angular errors of standard deviation $\sigma_\theta = 0.01745$ radians $\simeq 1°$. The density

of false measurments is $\rho = 10.0\ \text{km}^{-1}\ \text{radians}^{-1}$, so that on average $18\ \pi/2\ \rho = 282.7$ false measurements per scan are generated. The sector is scanned every second and the probability of detecting the target is $P_D = 1$. In Fig 6.2 the surveillance sector is divided into 54 cells of angular extent $10\sigma_\theta$ and of radial extent $100\sigma_r$, and the average (over 100 scans) number of false measurements per scan falling within each cell is shown. As expected the sample mean fluctuates about $1000\sigma_r\sigma_\theta\ \rho \simeq 5.236$. Initial estimates of target position and velocity, which are available to the filters, are in Cartesian co-ordinates. The standard deviation of the position error is 0.1 km on each co-ordinate and the standard deviation of the velocity error is $0.03\ \text{km sec}^{-1}$ for each co-ordinate. These principal problem parameters are listed in Table 6.1.

No direct correspondence between the parameters of this problem and the assessment example of section 5.2 (with Cartesian measurements) is possible. However the number of false measurements falling within a cell defined by the standard deviation of the true measurement error is $\rho\ \sigma_r\sigma_\theta \simeq 0.0054$ which corresponds to the parameter ρr of the assessment example. Also the non-dimensional parameter:

$$\frac{q\Delta t^4}{\sigma_r\ \sigma_\theta * \ \text{range}}$$

is analagous to $q\Delta t^4/r$ of the assessment example. Hence taking the standard range to be 11 km, *ie* to the centre of the sector, the equivalent parameters of the assessment example are approximately:

$$\rho r = 0.005$$

and

$$\frac{q\Delta t^4}{r} = 0.43 \quad .$$

The closest data point for which an estimate of track lifetime N_{AVE} is available for the assessment example is:

$$\rho r = 0.005$$

and

$$\frac{q\Delta t^4}{r} = 1$$

(see Fig 5.3). For these parameters:

$$N_{AVE} = 78.03 \qquad \text{for the PDAF}$$

and

$$N_{AVE} = 835.65 \qquad \text{for the CAF} \; .$$

Assuming an exponential distribution for track lifetime, the probability of a track surviving for at least t time steps is:

$$\exp(-\, t/N_{AVE}) \quad .$$

As noted, the average time for a target to pass through the sector is $t = 48$ seconds , therefore we can expect the PDAF to maintain track on about 54% of targets and the CAF to maintain track on about 94.4% of targets.

6.4 Simulation results

6.4.1 Correctly matched parameters

The CAF and the PDAF were applied to 100 replications of this problem for the standard parameters given above. For the PDAF 72% of the tracks were maintained while for the CAF 95% of the tracks were maintained. Thus the performance prediction of the previous paragraph was very accurate for the CAF but rather pessimistic for the PDAF. This discrepancy is probably due to the imprecise correspondence between the two problems and the neglect of any initial transient behaviour of the filters. Fig 6.3 shows an example of the CAF and the PDAF tracking a target across the sector. In this example the CAF successfully maintained track although the PDAF track became lost. Fig 6.4 shows an example of the extended Kalman filter tracking in the absence of false measurements. This figure shows the true measurements produced by the sensor; the increase in the measurement and tracking errors as the range from the sensor to the target increases can be clearly seen.

Fig 6.5 shows how the tracking performance of the CAF and the PDAF is affected by varying the density ρ of false measurements without changing the target trajectories or the true measurements. Tracking performance is shown for each of the 100 replications for ρ = 5, 10, 20, 30 and 40 km^{-1} $radians^{-1}$. For each of these values of ρ two traces are shown, one corresponding to the PDAF and the other corresponding to the CAF. Each trace has two levels H and L , according to whether a track was held or lost for each replication. It can be seen that for each value of ρ , every track held by the PDAF was also held by the CAF. One might expect that those tracks held by one of the filters for large ρ would also be maintained by that filter for smaller values of ρ . However this i. not always so,

because the random false measurements for a particular replication change completely as ρ is varied. Similarly, some tracks lost for small ρ are held for large ρ. As ρ is increased, the number of tracks maintained by each filter decreases.

This can be seen more clearly in Fig 6.6 where the percentage of tracks maintained by each filter is plotted against ρ. 95% confidence limites, derived from a binomial distribution since each replication is an independent Bernoulli trial, are given with each percentage. Also the average number of components before and after reduction are shown for the held tracks. The results exhibit similar trends as described in section 5.2 for the previous example. For small ρ, the PDAF and CAF both hold nearly all of the tracks, but for $\rho > 5\ \text{km}^{-1}\ \text{rad}^{-1}$, the CAF becomes more successful at maintaining track than the PDAF. The average number of mixture components generated increases with ρ, as does the required processing time recorded in Table 6.2 (part I). In this table the average computation time per step is given for held tracks and lost tracks separately. For small ρ, the computation time for held and lost tracks is similar although for large ρ the average timings for lost tracks are much greater, particularly for the CAF. This is due to the proliferation of feasible tracks which occurs for large ρ when the target is lost.

Also shown in Table 6.2 (part I) is an indication of the accuracy of the filters' own assessment of their tracking error in both position and velocity. This consistency measure, denoted $\bar{E}$, is derived as follows. At each time step, the quadratic form:

$$(\underline{x} - \hat{\underline{x}})^T P^{-1} (\underline{x} - \hat{\underline{x}}) \tag{6.7}$$

is evaluated, where $\underline{x}$ is the true value of the state vector, $\hat{\underline{x}}$ is the filter's estimate of the state vecor (*ie* the mean of the mixture distribution) and P is the overall covariance matrix of the mixture. If the filter's internal covariance P is compatible with the actual tracking error $\underline{x} - \hat{\underline{x}}$, then the expected value of the quadratic form (6.7) is 4, because for this tracking problem the state vector is four-dimensional. The statistic $\bar{E}$ given in Table 6.2 is calculated by averaging (6.7) over all time steps for held and lost tracks separately. Since the tracking error $\underline{x} - \hat{\underline{x}}$ may be correlated over several time steps and it may not be a Guassian variable, we cannot expect the distribution of the sum of (6.7) over all time steps to have a χ^2 distribution. However since $\bar{E}$ is usually the result of an average over many hundreds of time steps, if $\bar{E}$ deviates from 4 by as much as one unit, it is reasonable to conclude that the filter's internal covariance P is incompatible with the actual tracking error. Table 6.2 shows that for both filters the value of $\bar{E}$ for maintained tracks is usually slightly less than 4, but within 10% of this figure. This indicates that the achieved tracking error is a little better than the filters' assessment, and this is possibly because the maintained tracks are a biased sample in favour of the more accurate tracks. For lost tracks, $\bar{E}$ is usually very much larger than 4, showing that the filters seriously underestimate the tracking error. The CAF is worse than the PDAF in this respect.

The actual mean square position tracking errors achieved by the filters for the first 20 time steps are shown in Fig 6.7 for ρ = 5, 10 20 and 40 km^{-1} rad^{-1} . These results are obtained by averaging the square of the position error at a particular time step over all replications. The mean square error is also shown for the maintained tracks only. As a reference level the tracking error for the

extended Kalman filter, which is supplied only with true measurements, has been plotted. The results shown in Fig 6.7 are intuitively reasonable. The Kalman filter tracking errors are smaller than those for the filters which have to cope with false measurements, and the errors averaged over all tracks are greater than those for held tracks only. Errors for the CAF tend to be smaller than those of the PDAF, although for $\rho = 20\ \mathrm{km}^{-1}\ \mathrm{rad}^{-1}$ the 13 tracks held by the PDAF have a smaller mean square position error than the 86 tracks held by the CAF. Thus it appears that PDAF tracks are only able to survive in this case if they are able to achieve a relatively small position error in the early stages of the track (average track length being 48 time steps).

6.4.2 Mismatched model noise

One would expect performance to degrade if the assumed values of the filter parameters ρ, q, σ_r, σ_θ and P_D differ from their correct values. Here we examine the effect of a mismatch in the parameter q, the variance of the model noise, which describes the manoeuvrability of the target. If the values of q assumed by the filters is less than the correct value, the filters may judge actual target manoeuvres to be highly improbable, in which case true measurements may be rejected or given a very low probability weighting. If the value of q is too high, the filters may give too much weighting to false measurements which could only be true if the target had performed a large manoeuvre incompatible with the correct value of q. An adaptive version of the PDAF which learns an unknown value of q from a set of possible candidates has been proposed by Gauvrit[51]. However for the present study only the fixed parameter filter has been considered.

The 100 trajectories simulated for the standard problem parameters (see Table 6.1) were used to investigate the effect of supplying the

filters with the incorrect value of q . The CAF and the PDAF have been applied with $\sqrt{q}$ set to 0.01, 0.025, 0.1 and 0.25 km sec^{-2}, as well as the correct value of 0.05 km sec^{-2}. The percentage of tracks maintained for these values are given in Fig 6.8 together with the average number of components generated for the maintained tracks. (The error reference for the track loss criterion is obtained from the Kalman filter using the correct value of $\sqrt{q}$.) It appears that the CAF performance is less sensitive to parameter mismatch than the PDAF. This extra flexibility of the CAF is due to the filter's ability to retain several feasible tracks. Indeed there is a slight (probably insignificant) performance improvement for the CAF when $\sqrt{q}$ is doubled, although the percentage of tracks held by the PDAF is reduced from 72% to 16%. When $\sqrt{q}$ is increased to five times its correct value, the number of tracks held by the CAF is reduced by about one third, although the PDAF now loses all of the tracks. As $\sqrt{q}$ is decreased from its correct value, the performance of both filters degrades at a similar rate. Also note that the number of components generated by the filter increases with $\sqrt{q}$. This is because with increasing $\sqrt{q}$, the filters believe the target to be capable of larger manoeuvres and so are more ready to accept false measurements.

Average computation time per step and the error statistic $\bar{E}$ are given in part II of Table 6.2. CAF processing time increases with $\sqrt{q}$ due to the increasing number of components generated. The reasonable CAF track maintainance performance obtained when $\sqrt{q}$ is five times its correct value is at the expense of a 60 fold increase in computation time for held tracks. The PDAF incurs only a small increase in processing time although performance falls off rapidly for q too large.

The use of an incorrect value for q has a noticeable effect on the error statistic $\bar{\bar{E}}$. When q is too large, $\bar{\bar{E}}$ is significantly less than four for both held and lost tracks, showing that the filters are overestimating their tracking errors, so that the filter gains are set too high. When q is too small, $\bar{\bar{E}}$ is much greater than four showing that the filters are overoptimistic about their tracking performance. In this case the filter gains are too small so that the filters are insufficiently responsive to received measurements. The actual mean square position errors for maintained CAF tracks with mismatched q are shown in Fig 6.9 for the first twenty time steps. After the first few time steps, there is a clear trend for tracking error to increase as the assumed value of q deviates further from its correct value. When $\sqrt{q}$ is five times or one fifth of its correct value, the mean square position error after the tenth time step is approximately ten times that obtained with the correct value of q.

6.4.3 Trajectories with deliberate manoeuvres

In this section we investigate the tracking performance of the filters when the target executes deterministic manoeuvres which do not obey the filter model. This is a further degree of mismatch between the assumed and the actual target behaviour. Two types of trajectory have been simulated, both of which start with a constant velocity course. The initial position and velocity of the target on entering the sector is chosen as described in section 6.3. For the first type of trajectory, the target proceeds on the constant velocity course for 12 seconds after entering the sector, then performs a sinusoidal weave with half amplitude 1 km and frequency 0.05 Hz, and finally returns to a constant velocity course after 35 seconds of weaving. For this weave the maximum target acceleration is about 10'g' at the extremities of

the sinusoid. An example of this type of trajectory is shown in Fig 6.10. For the second type of trajectory, after having travelled in a straight line for 25 seconds, the target turns in a circular arc for 15 seconds and then resumes a constant velocity course. The radius of the arc is 1 km, so that for the mean target speed, the acceleration whilst turning is about 9'g'. An example of this type of trajectory is shown in Fig 6.11.

For these trajectories, the motion of the target switches between periods of constant velocity motion and periods of high 'g' manoeuvres. In these circumstances, one would ideally employ different target models for the two phases of the trajectory. For example, if the second order model were used, $q = 0$ would be correct for constant velocity motion while a value of $\sqrt{q}$ close to the maximum acceleration that can be achieved by the target might be appropriate (but not ideal) for periods of manoeuvre. Usually the filter does not know when the target is going to execute a manoeuvre and so adaptive tracking schemes have been suggested. For instance the Interacting Multiple Model (IMM) algorithm of Blom[40,52] assumes that the target motion may be described by one of a set of possible models, and that the motion changes abruptly between these models with some assumed switching probability. This introduces a further degree of uncertainty into the tracking problem which gives rise to a large increase in the number of components making up the mixture distribution of the target state. Houlès and Bar-Shalom[14] have applied the IMM algorithm with the PDAF to an example which is very similar to the sector scan problem. However for the present study a single target model with fixed parameters has been employed to avoid the added complication of a multiple model filter.

One hundred replications of each of the two types of trajectory have been generated together with measurements with the standard parameters (see Table 6.1). Figs 6.12 and 6.13 show the percentage of these tracks maintained by the CAF and the PDAF for different values of the assumed model noise standard deviation $\sqrt{q}$. The correct measurement parameters σ_r, σ_θ and ρ were supplied to the filters and as usual the reference error for the track loss criterion was obtained from the Kalman filter with $\sqrt{q} = 0.05$ km sec^{-2} . For both types of trajectory the performance of the PDAF is poor, with the percentage of held tracks rising above 10% only for $\sqrt{q} = 0.05$ km sec^{-2} . For the CAF the best performance is achieved at the higher value of $\sqrt{q} = 0.1$ km sec^{-2} , for which 97% of weaving tracks and 99% of circling tracks were held. Note that this value of $\sqrt{q}$ is close to the maximum acceleration of the targets when they are performing their manoeuvres. As in the case of second order model trajectories with mismatched q , the performance of the CAF appears to be less sensitive than the PDAF to variation of q ; reasonable CAF performance being obtained with $\sqrt{q} = 0.25$ and 0.05 km sec^{-2} . It would be interesting to see if use of the IMM algorithm would improve the PDAF performance.

As already indicated, a single value of q is a compromise for this problem. This is highlighted in Figs 6.14 and 6.15 which show the mean square position error as a function of the time step for $\sqrt{q}$ set to 0.025, 0.05, 0.1 and 0.25 km sec^{-2}. For maintained tracks, the minimum error for the initial constant velocity path is obtained for the smallest value of q , although the minimum tracking error during the manoeuvre is obtained for $\sqrt{q} = 0.1$ km sec^{-2} . Generally for fixed q the tracking error is greatest during the target manoeuvre, and this is when tracks are usually lost, as can be seen from the traces showing

error averaged over all tracks. However for the high value of $\sqrt{q} = 0.25$ km sec^{-2}, the error for maintained tracks is fairly constant over the whole trajectory after the initial transient. For the weaving trajectories with $\sqrt{q} = 0.1$ km sec^{-2} (for which the CAF performs best), it can be seen from Fig 6.14 that the largest CAF tracking errors occur just after the turning points of the weave, when the target is pulling maximum 'g'. This is also clear in the tracking example shown in Fig 6.10.

The average processing time per step and the error statistic $\bar{E}$ for these simulations are recorded in Table 6.3. These values are averaged over all time steps of the trajectories, including periods of manoeuvre and constant velocity motion. As for the case of mismatched q with second order model trajectories, the processing time for the CAF rises with q, at first gently and then steeply for $\sqrt{q} > 0.1$ km sec^{-2}. This is reflected in the number of mixture components generated (see Figs 6.12 and 6.13). PDAF computation time also rises with q, but does not show the sharp rise of the CAF for $\sqrt{q} > 0.1$ km sec^{-2}. For $\sqrt{q} \leq 0.1$ km sec^{-2}, average CAF processing time is three to four times greater than that of the PDAF.

Since the generated trajectories do not match the filter's target model and the level of manoeuvre changes in mid-course, we cannot expect $\bar{E}$ to be very close to four. However for $\sqrt{q} = 0.1$ km sec^{-2}, when the CAF performs best, the values of $\bar{E}$ for weaving and circular manoeuving targets are within an order of magnitude of four, which suggests that this value of q is a reasonable compromise for these trajectories.

6.5 Conclusions

The sector scan problem presented in this chapter provides a more realistic demonstration of the baseline problem. The extended Kalman filter has been employed to manage the non-linear relationship between the measurements in polar co-ordinates and the target model in Cartesian co-ordinates. Essentially the measurement association and evaluation of the probability weights of the mixture pdf are performed in polar co-ordinates, while the calculation of the mean and covariance of each mixture component (the filtering operation) is performed in Cartesian co-ordinates.

The effect on filter performance of a mismatch between the statistics of the actual target trajectory and the assumed filter model has been studied. For trajectories generated by the second order model, CAF performance is less sensitive to mismatch than the PDAF. Also for the deterministic manoeuvres, the CAF achieves acceptable performance over a wider range of filter model parameters than the PDAF. This extra flexibility of the CAF is due to the filter's ability to retain several feasible tracks. As might be expected, statistical anlaysis shows that the filters' internal assessment of tracking error is unreliable if the filter model is incorrect. Filter assessment is optimistic when the manoeuvre parameter q is too small and it is pessimistic when q is too large.

Table 6.1

PRINCIPAL PROBLEM PARAMETERS FOR THE SECTOR SCAN PROBLEM

Surveillance sector is the region:

$$X > 0 \text{ km} , \; Y > 0 \text{ km}$$

and

$$2 \text{ km} < \sqrt{X^2 + Y^2} < 20 \text{ km} .$$

Second order target model:

Standard deviation of acceleration noise for each co-ordinate is:

$$\sqrt{q} = 0.05 \text{ km sec}^{-2} \simeq 5\text{'g'} .$$

Initial target speed (on entering the sector) is drawn from a Gaussian distribution with mean 0.3 km sec^{-1} and standard deviation 0.02 km sec^{-1}.

Initial estimate of target state supplied to filters is a Gaussian perturbation about the true state. For each Cartesian co-ordinate, standard deviation of velocity error is 0.03 km sec^{-1} and standard deviation of position error is 0.1 km.

True measurements have a Gaussian range error with standard deviation σ_r = 0.03 km and a Gaussian bearing error with σ_θ = 0.01745 radians $\simeq 1°$.

Probability of detection $P_D = 1$.

False measurements are uniformly distributed over the surveillance sector in polar co-ordinates with density ρ = 10.0 km^{-1} radian^{-1} .

Table 6.2

PROCESSOR TIMINGS AND ERROR STATISTIC $\bar{E}$ FOR SECTOR SCAN PROBLEM WITH TRAJECTORIES GENERATED BY THE SECOND ORDER MODEL

	Parameters			Tracks held or lost	Average cpu time for single step (ms)		Error statistic $\bar{E}$	
	Density of false measurements ρ (km^{-1} rad^{-1})	Actual trajectory noise $\sqrt{q}$ (km sec^{-2})	Assumed filter model noise $\sqrt{q}$ (km sec^{-2})		CAF	PDAF	CAF	PDAF
I Correctly Matched Parameters (Section 6.4.1)	0.5	0.05	0.05	H	0.321	0.148	3.975	3.980
				L	-	-	-	-
	1.0	0.05	0.05	H	0.352	0.165	3.938	3.942
				L	0.347*	0.149*	2770.0*	5823.0*
	5.0	0.05	0.05	H	0.762	0.300	3.976	4.128
				L	0.675*	0.294	24.41*	2101.000
	10.0	0.05	0.05	H	1.650	0.486	3.995	3.859
				L	1.930	0.591	1825.000	2368.000
	20.0	0.05	0.05	H	5.400	0.810	4.097	3.096
				L	5.090	1.710	2667.000	197.700
	30.0	0.05	0.05	H	15.000	1.13*	3.660	3.124*
				L	52.800	2.72	2081.000	6.694
	40.0	0.05	0.05	H	68.400	1.48*	3.638	3.142*
				L	822.000	3.95	539.000	5.457

* Indicates a small sample (less than five replications)

Table 6.2 (concluded)

	Parameters			Tracks held or lost	Average cpu time for single step (ms)		Error statistic $\bar{E}$	
	Density of false measurements ρ (km^{-1} rad^{-1})	Actual trajectory noise $\sqrt{q}$ (km sec^{-2})	Assumed filter model noise $\sqrt{q}$ (km sec^{-2})		CAF	PDAF	CAF	PDAF
II Mismatched Model Noise (Section 6.4.2)	10.0	0.05	0.01	H	1.20	0.452*	314.200	39.24*
				L	1.18	0.425	12220.000	12260.000
	10.0	0.05	0.025	H	1.45	0.457	10.280	11.370
				L	1.34	0.449	12200.000	5094.000
	10.0	0.05	0.05	H	1.65	0.468	3.995	3.859
				L	1.93	0.591	1825.000	2368.000
	10.0	0.05	0.1	H	2.38	0.486	2.254	2.017
				L	2.26*	1.010	1.806*	2.966
	10.0	0.05	0.25	H	98.90	-	0.995	-
				L	281.00	1.480	1.318	2.977

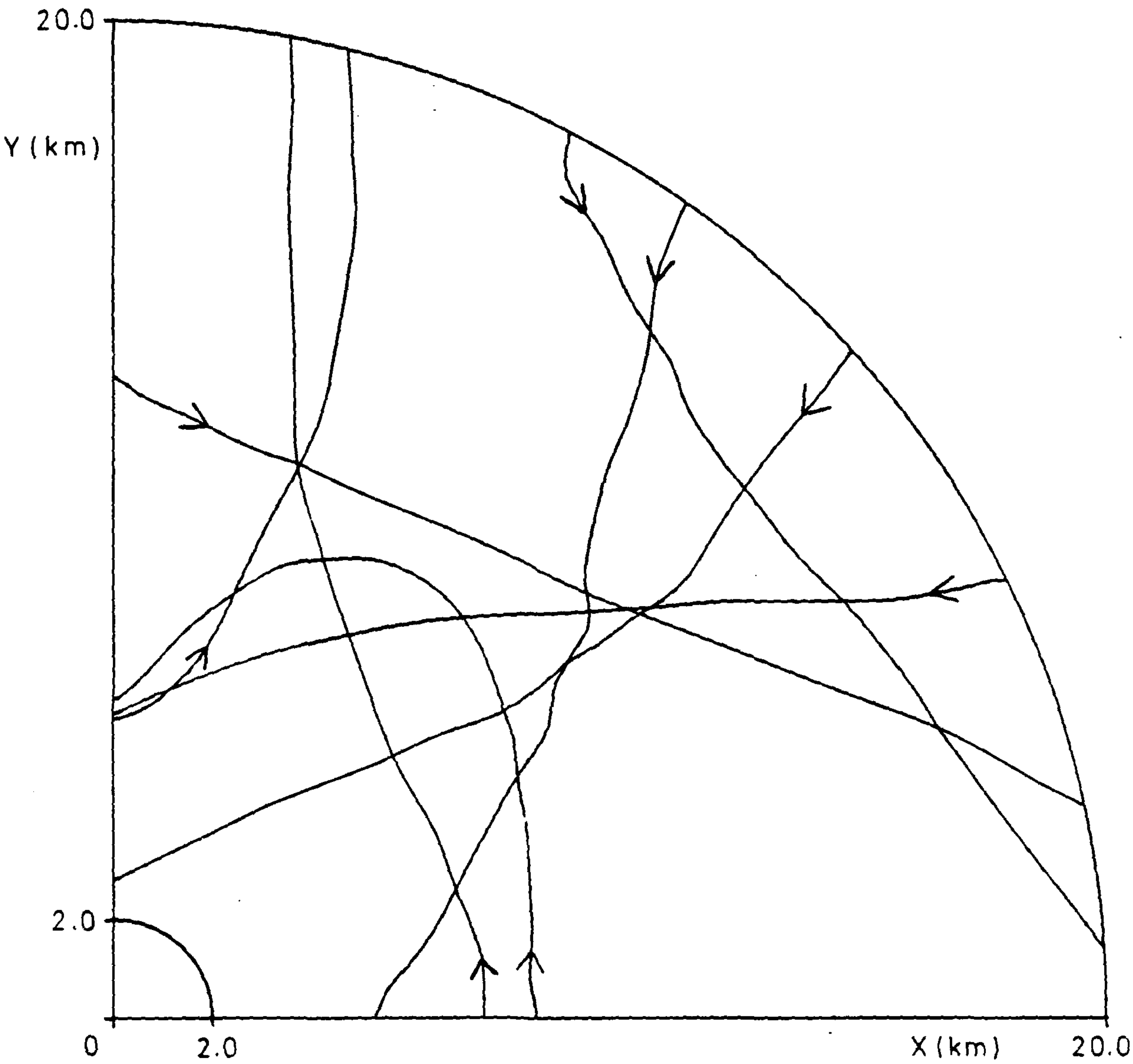
* Indiates a small sample (less than five replications)

Table 6.3

PROCESSOR TIMINGS AND ERROR STATISTIC $\bar{E}$ SECTOR SCAN PROBLEM WITH DETERMINISTIC TRAJECTORIES

	Assumed filter model noise $\sqrt{q}$ (km sec^{-2})	Tracks held or lost	Average cpu time for single step (ms)		Error statistics $\bar{E}$	
			CAF	PDAF	CAF	PDAF
Weave Manoeuvre	0.010	H	1.31*	0.435*	989.8*	505.6*
		L	1.22	0.429	5922.0	7371.0
	0.025	H	1.69	0.514	328.0	184.00
		L	1.36	0.469	6110.0	5821.70
	0.050	H	1.77	0.478	17.95	65.70
		L	1.78	0.645	4075.00	1825.00
	0.100	H	2.57	0.512*	2.366	2.752*
		L	1.87*	1.140	2372.0*	3.312
	0.250	H	113.00	-	0.9630	-
		L	221.00	1.580	1.0800	2.430
	0.300	H	507.00	-	0.6860	-
		L	744.00	1.620	1.4750	2.386
Circular Arc Manoeuvre	0.025	H	1.48*	0.463*	130.6*	41.9*
		L	1.38	0.478	19720.0	15230.0
	0.050	H	1.70	0.487	56.87	626.7
		L	3.73	0.534	7369.00	4899.0
	0.100	H	2.43	0.495*	25.05	15.06*
		L	2.11*	1.150	528.2*	12.39
	0.250	H	101.50	-	4.922	-
		L	305.40	1.610	2.932	3.807
	0.300	H	307.00	-	2.701	-
		L	834.00	1.640	2.554	3.713

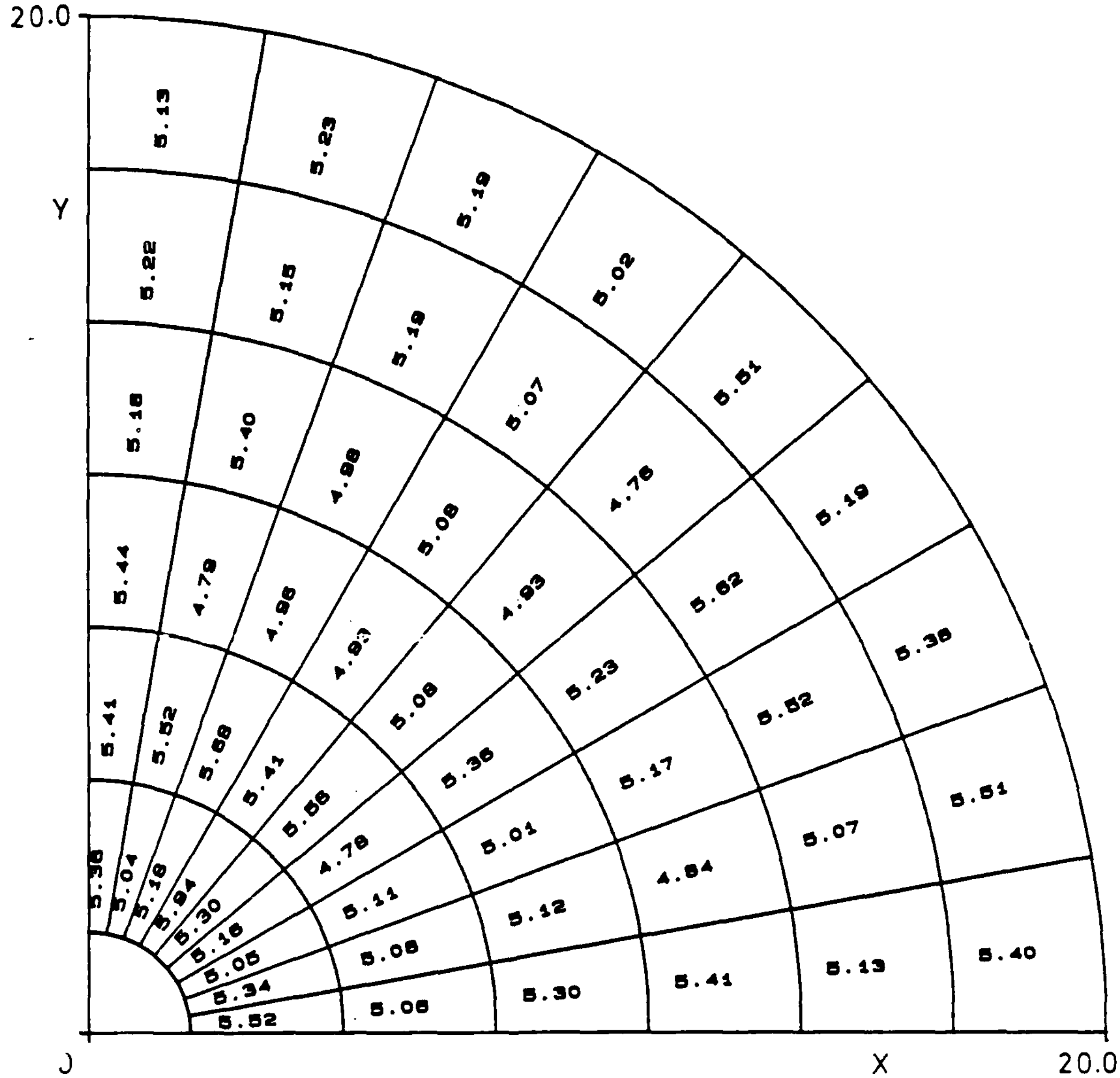
* Indicates a small sample (less than five replications)



Initial target speed : mean 0.3 km s^{-1}
standard deviation 0.02 km s^{-1}

Acceleration noise standard deviation :
$\sqrt{q} = 0.05$ km $s^{-2} \triangleq 5$ 'g'

Fig 6.1 A sample of eight target trajectories from the sector scan problem (note that targets pass through the sector one at a time)



False measurement density ρ = 10 km^{-1} radian^{-1}

Fig 6.2 Average number of false measurements falling within cells

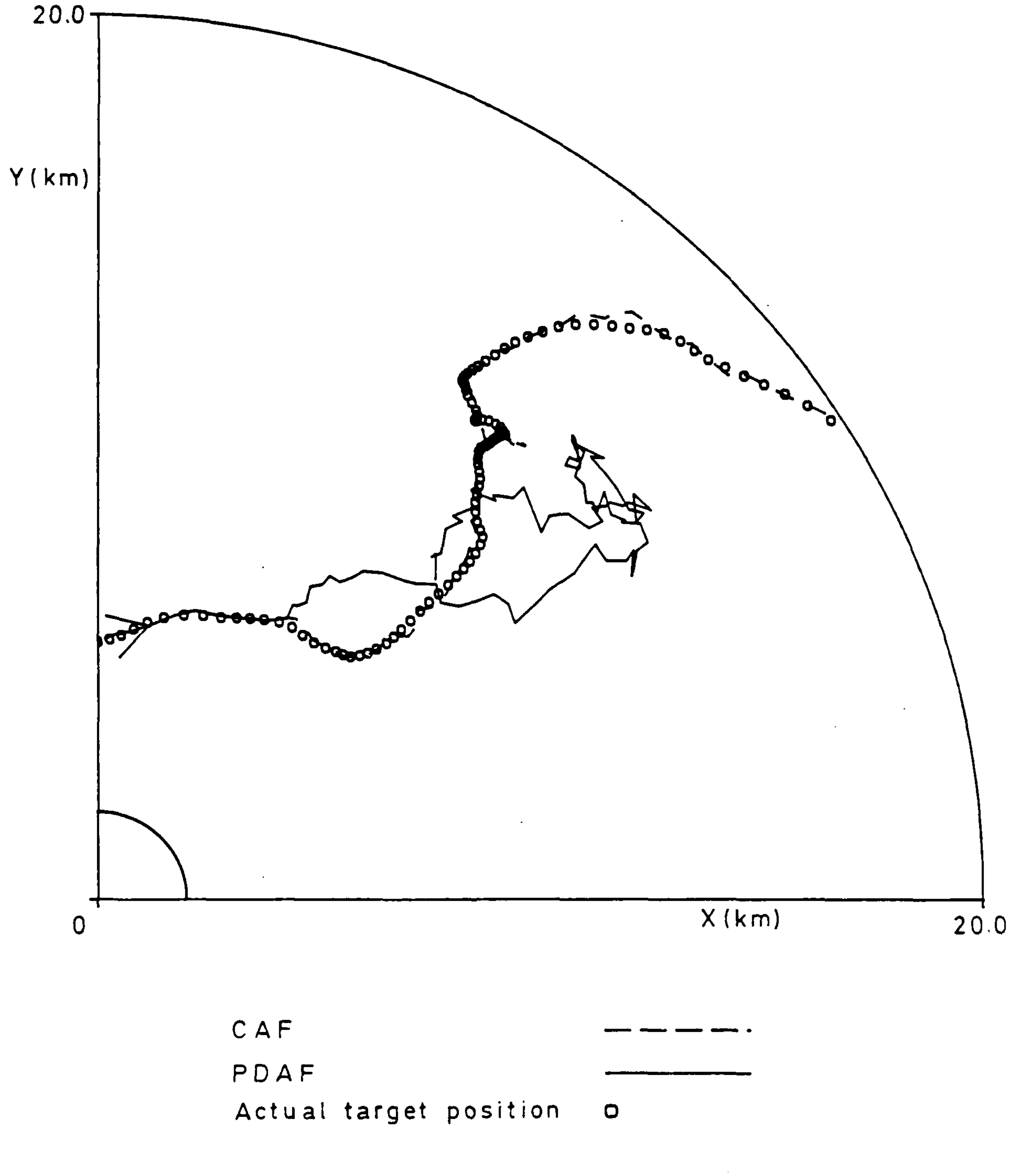


Fig 6.3 An example of CAF and PDAF tracking for the sector scan problem (parameters of Table 6.1)

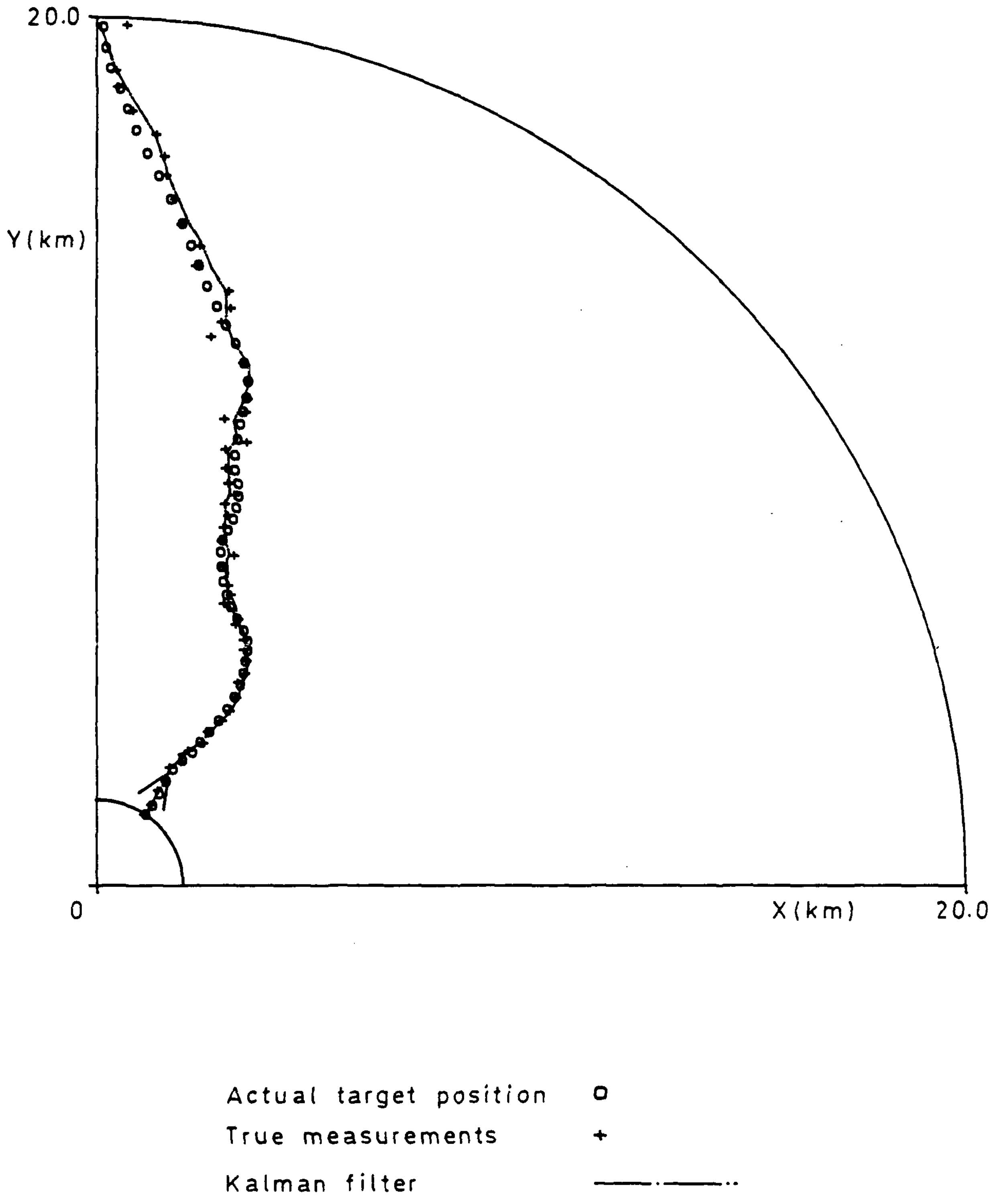


Fig 6.4 An example of tracking with the extended Kalman filter (parameters of Table 6.1, but ρ = 0)

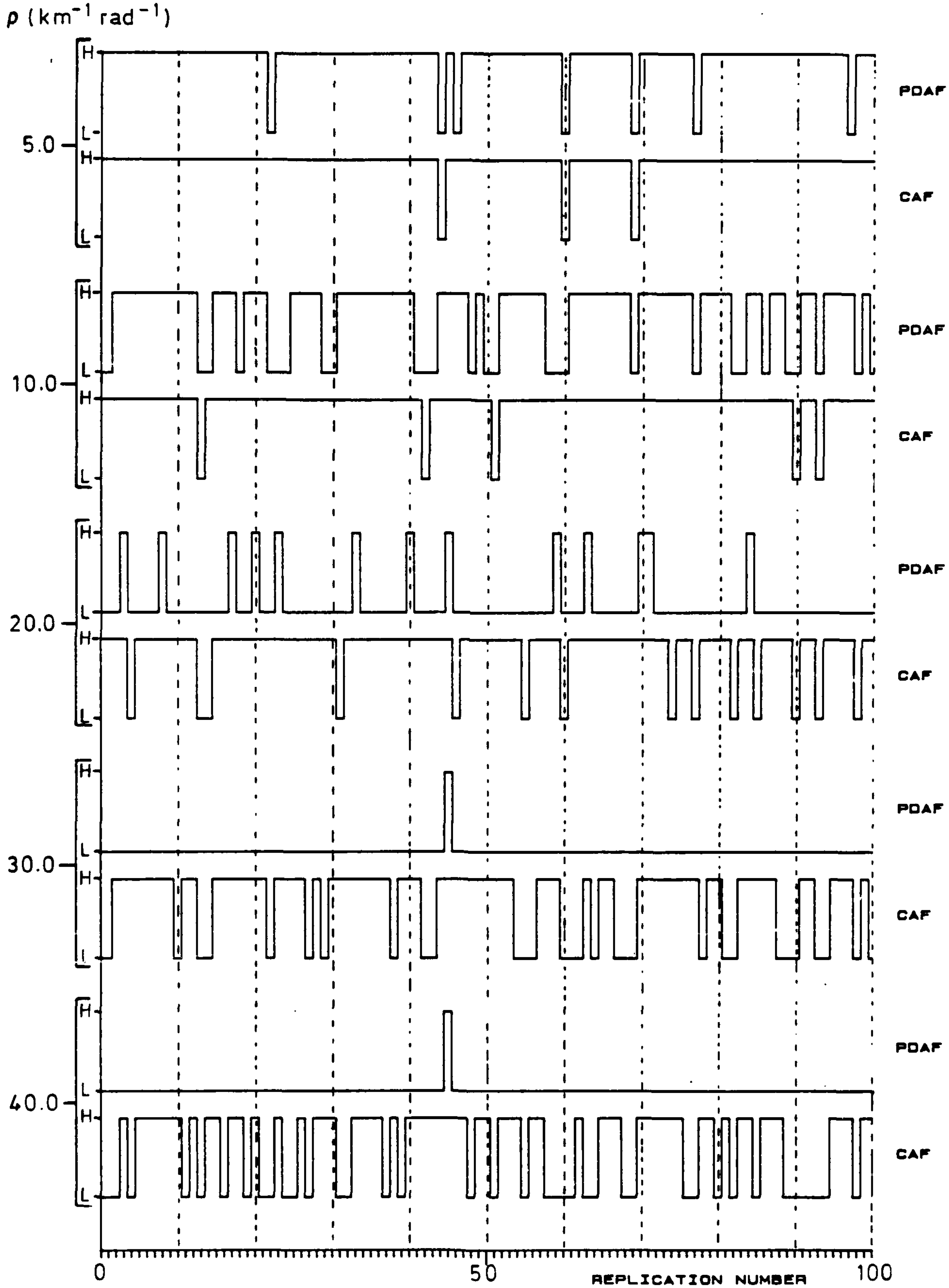


Fig 6.5 Filter performance at each replication for increasing density ρ of false measurements

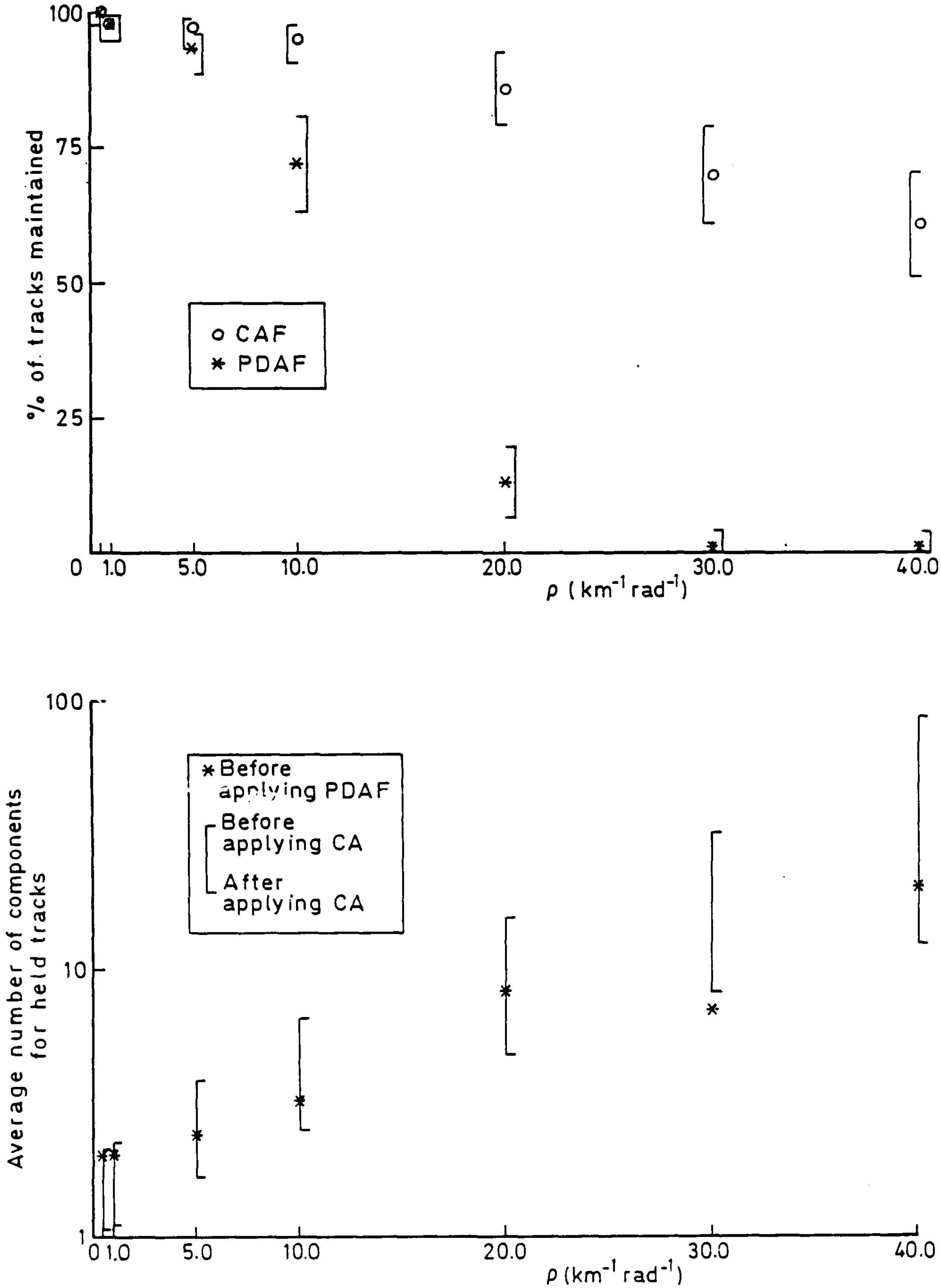


Fig 6.6 Performance of CAF and PDAF for sector scan problem as a function of ρ (other parameters from Table 6.1)

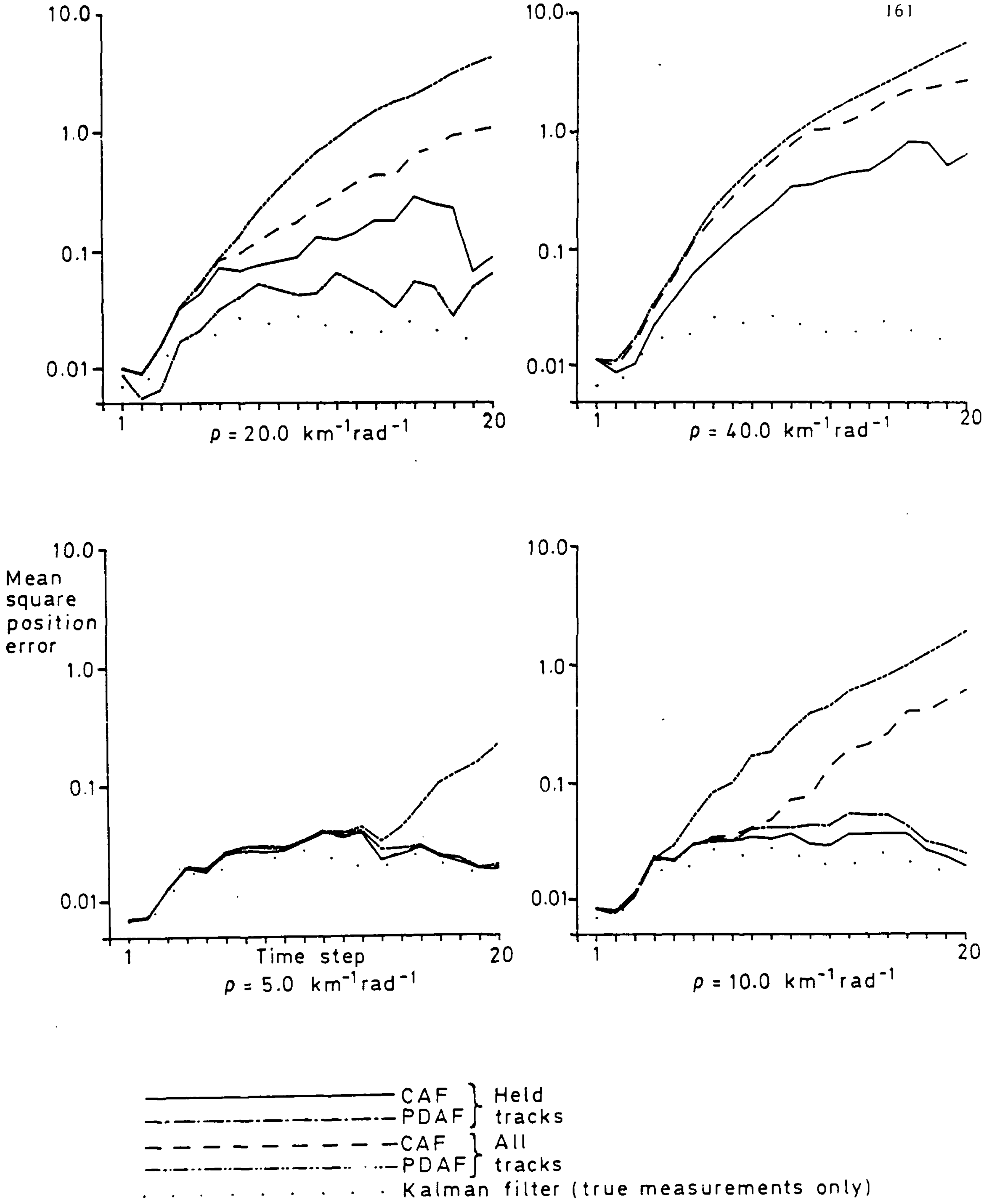


Fig 6.7 Achieved mean square position error

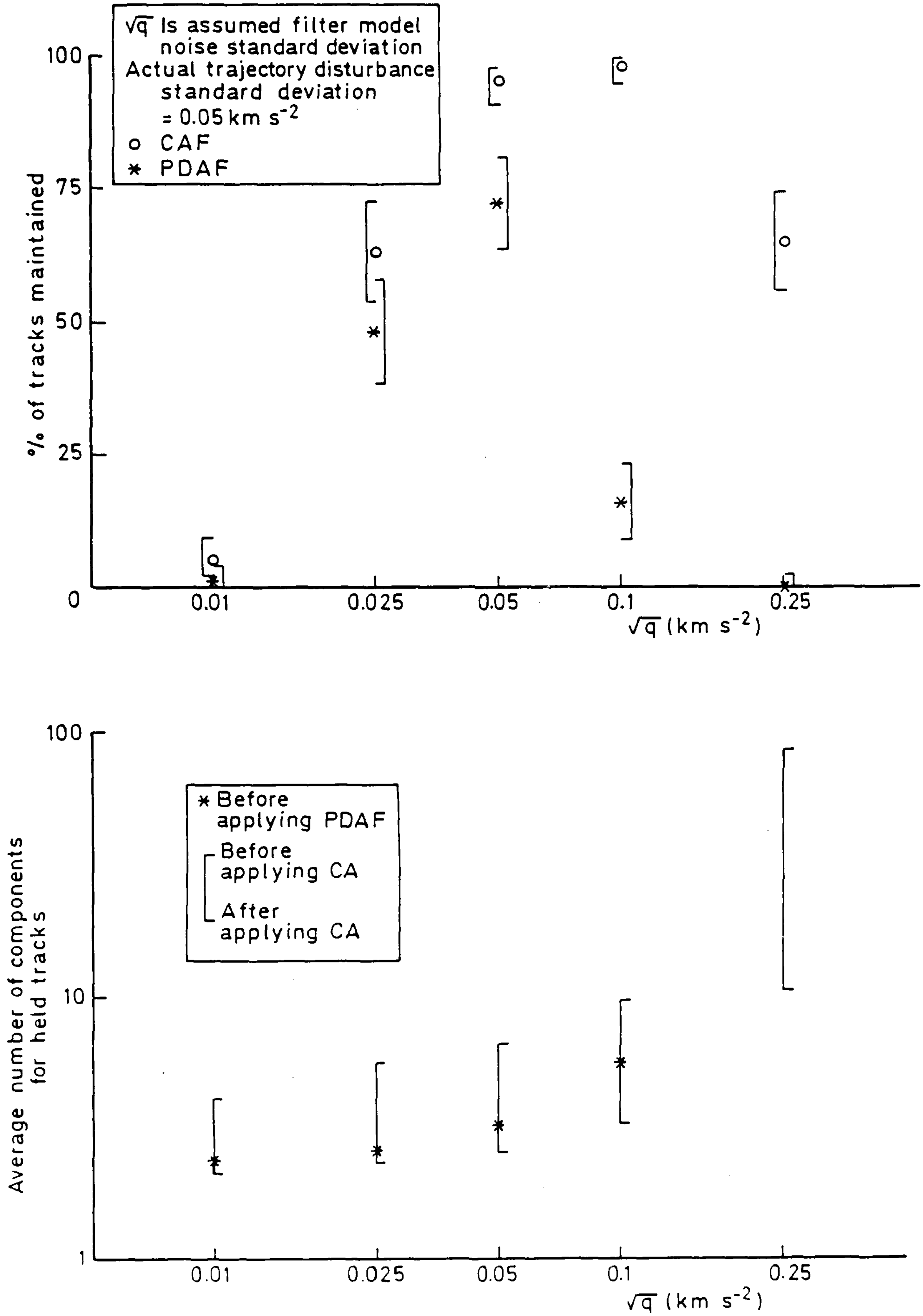
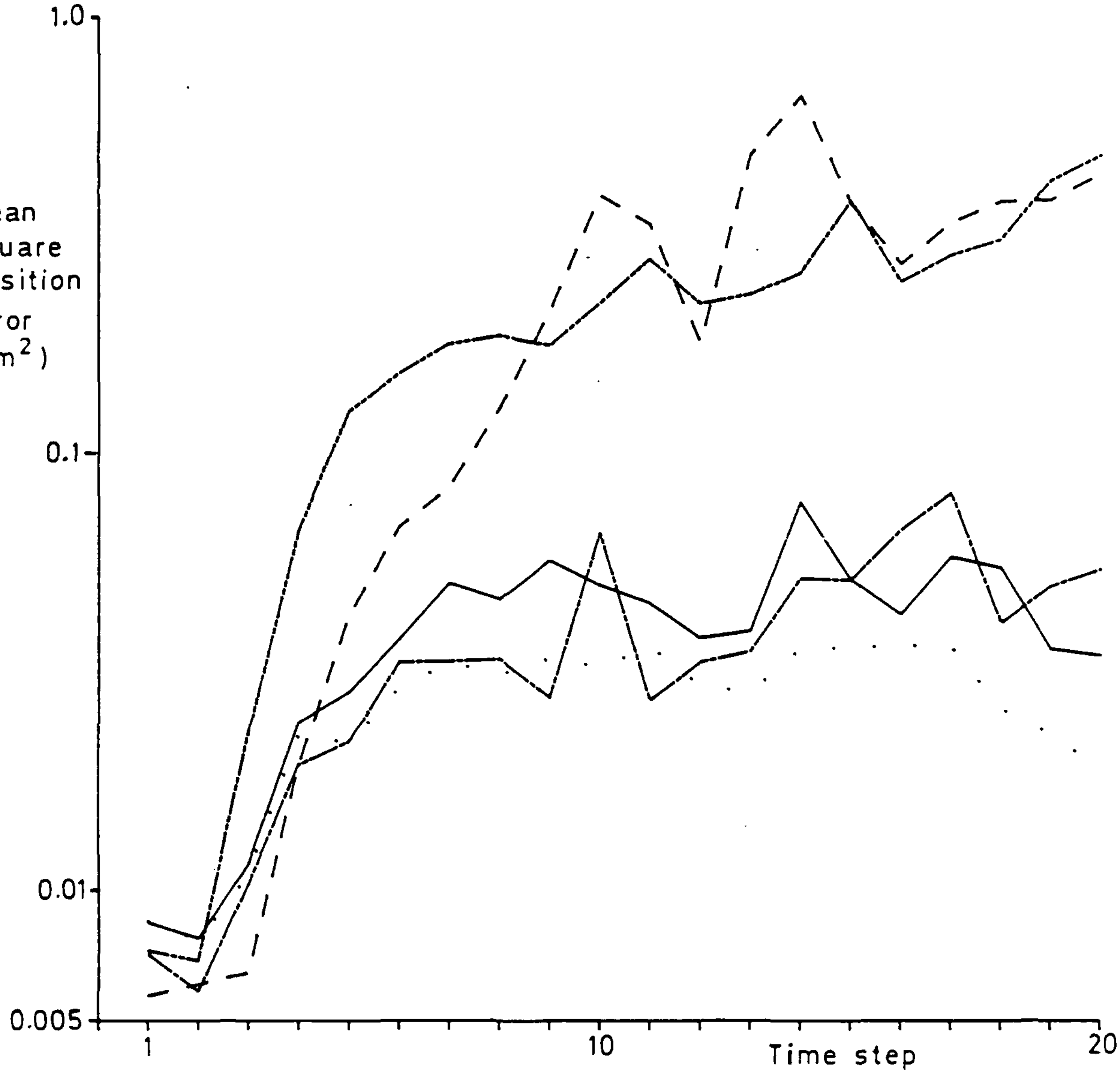


Fig 6.8 Filter performance for mismatched q



Assumed filter model noise standard deviation

—·—··—·—··— $\sqrt{q} = 0.25$ km s^{-2}

———————— $\sqrt{q} = 0.1$ km s^{-2}

· · · · · · · · · $\sqrt{q} = 0.05$ km s^{-2} = Actual value for trajectory disturbance

—·—·—·—·— $\sqrt{q} = 0.025$ km s^{-2}

— — — — — — $\sqrt{q} = 0.01$ km s^{-2}

Fig 6.9 Mean square position error for maintained CAF tracks with mismatched q

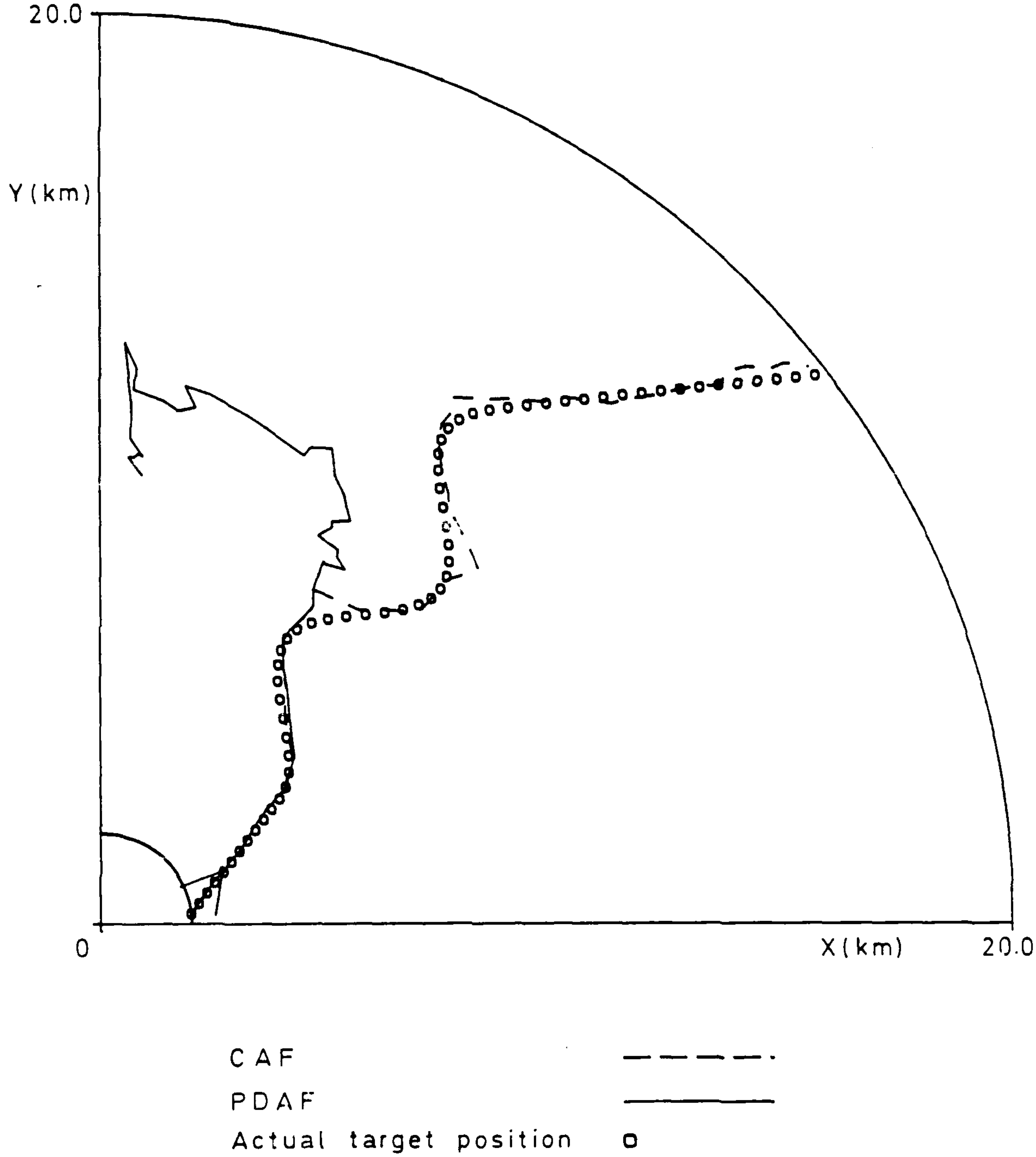


Fig 6.10 An example of tracking a weaving target

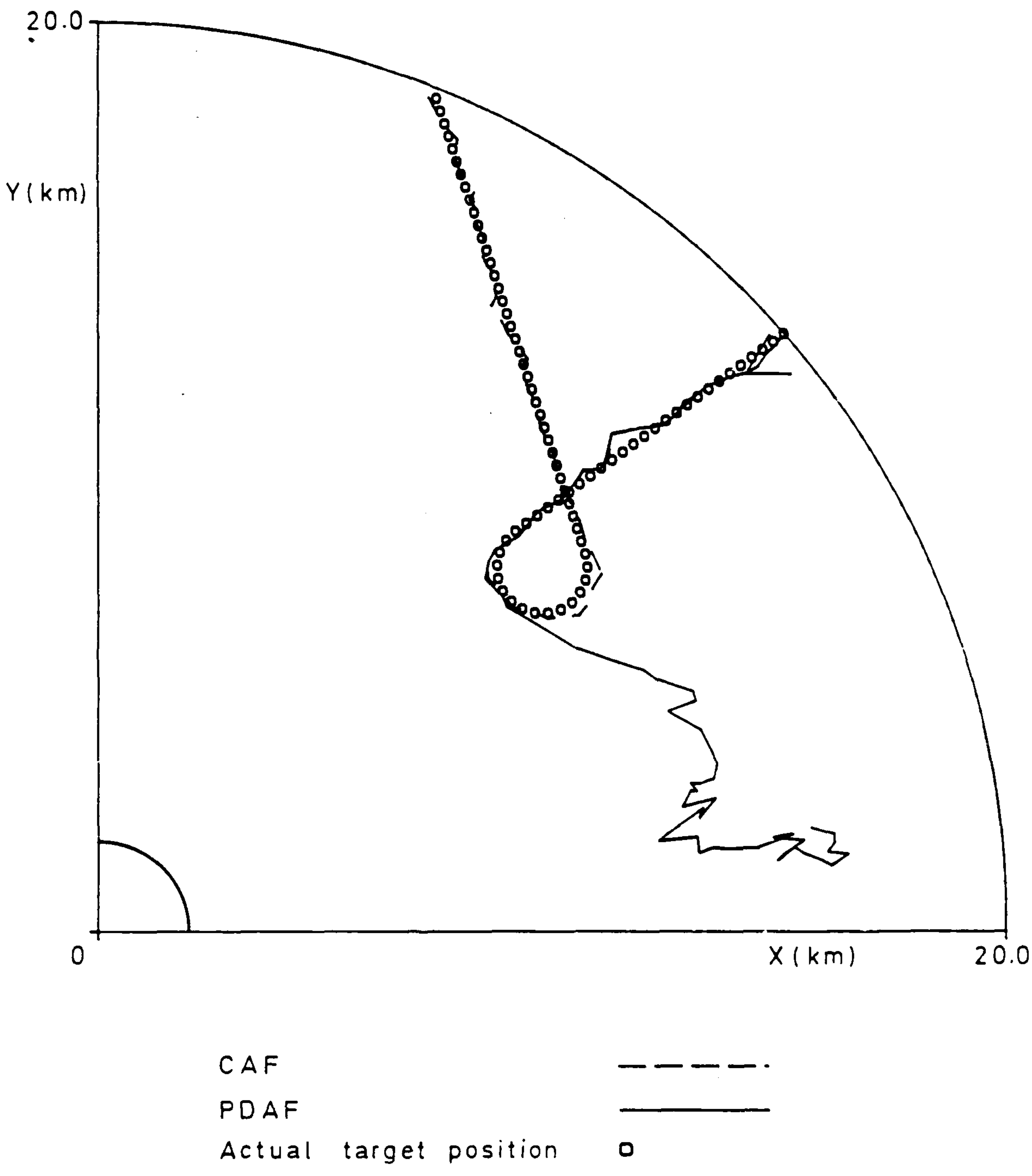
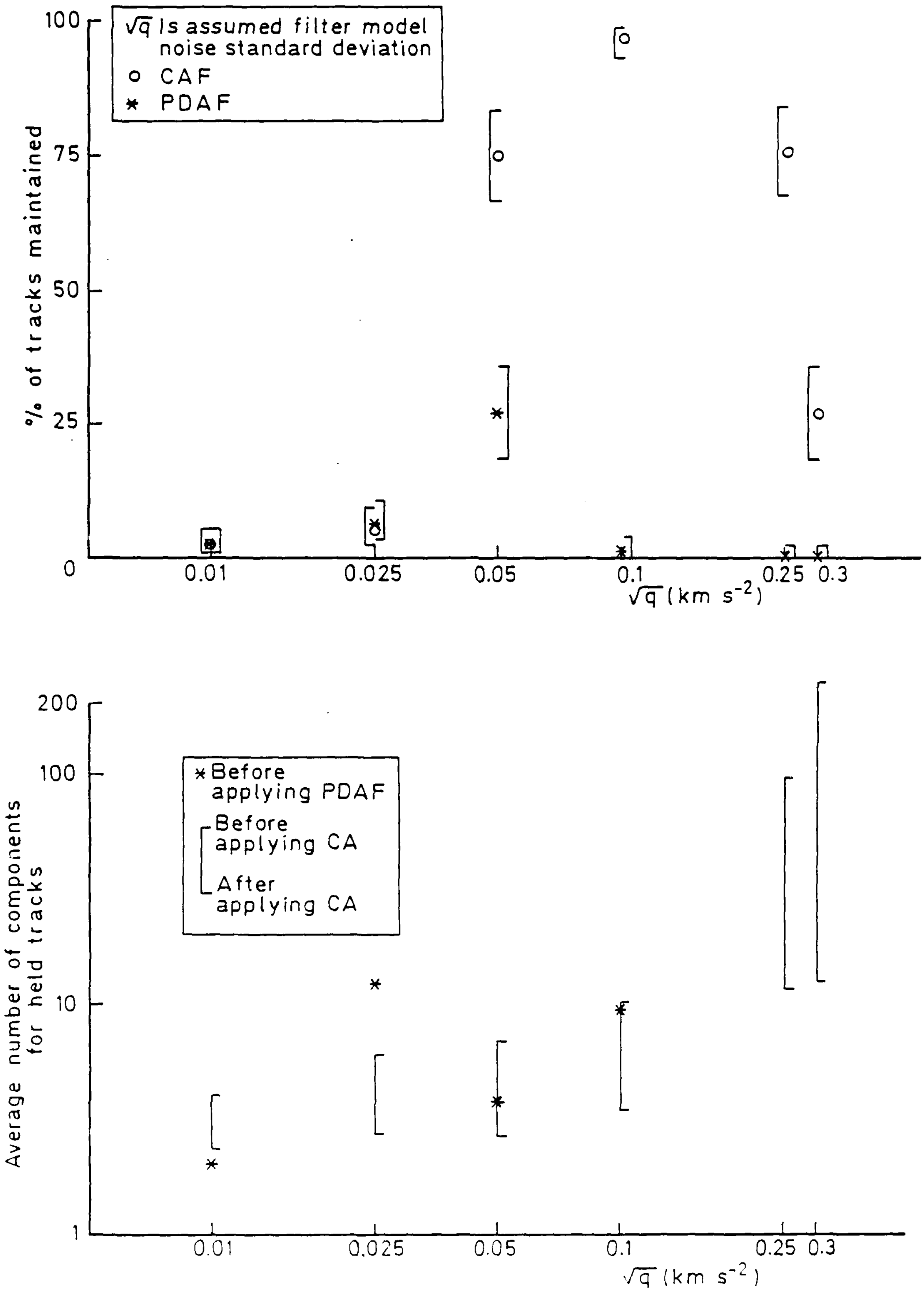
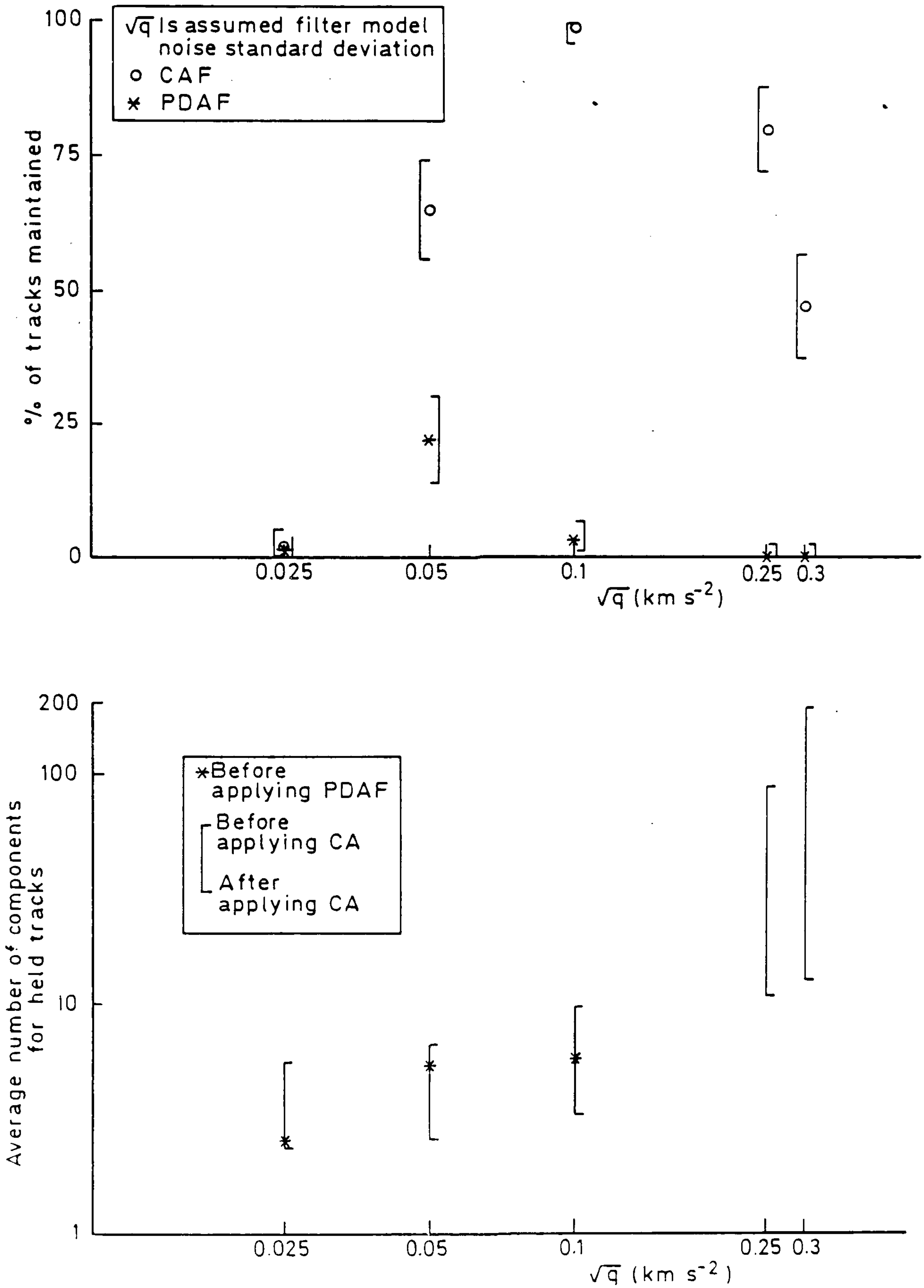


Fig 6.11 An example of tracking a target which executes a turning manoeuvre



[illegible] 6.1[illegible] Filter performance for a target which executes a weaving manoeuvre



Fi[illegible].13 Filter performance for a target which executes a turning manoeuvre

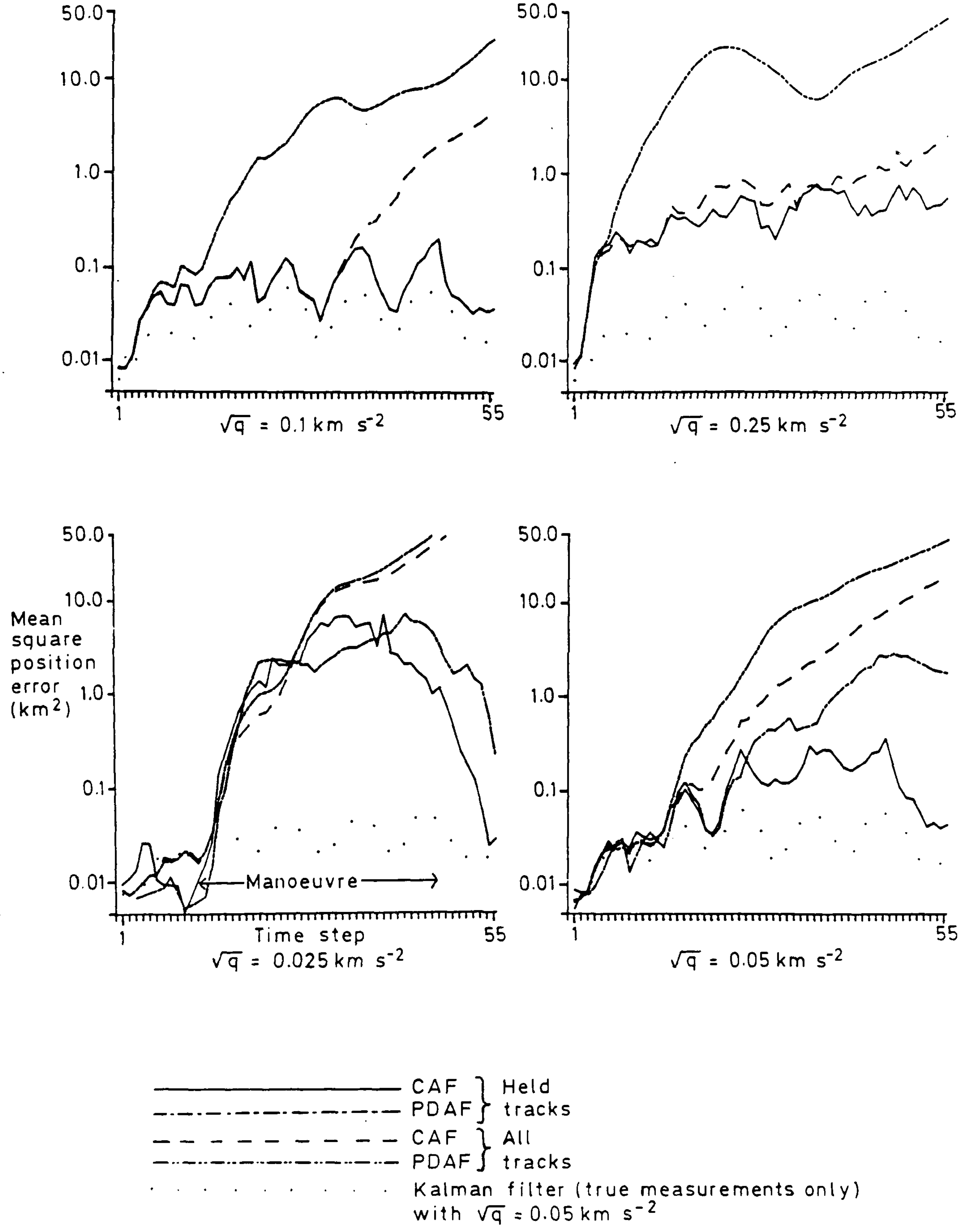


Fig 6.14 Mean square position error for a target executing a weaving manoeuvre

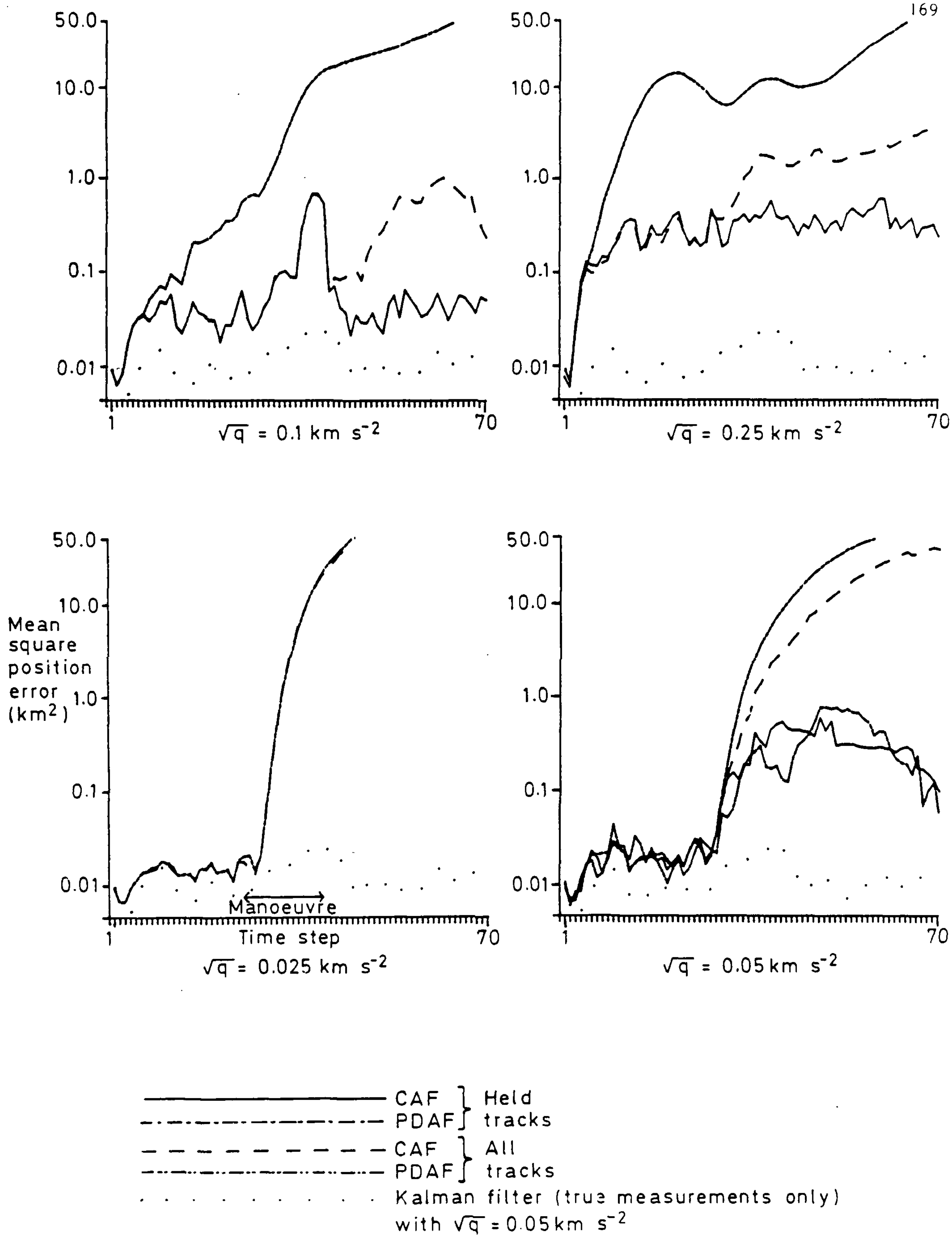


Fig 6.15 Mean square position error for a target executing a turning manoeuvre

7 THE DATA FUSION PROBLEM

7.1 Introduction

In this chapter the problem of fusing information from a number of sources is considered. For the baseline problem, the only available data are the position measurements received from a single sensor at each time step. In many practical cases an imperfect classification of these position measurements (into true or false categories) may be available. A simple extension of the baseline filter enables this classification information to be incorporated into the posterior pdf of target state (section 7.2). It is also possible that several independent sensors may be available to supply position measurements at each time step. Data from each sensor may be incorporated sequentially (section 7.3), although this may be time consuming. In section 7.4 we derive a computationally efficient suboptimal filter for combining information from a primary sensor with measurements from an auxiliary sensor. In the example considered, the auxiliary sensor gives only bearing information but does include an imperfect classification of these measurements. The sub-optimal filter uses the auxiliary measurements to modify only the probability weights of the mixture distribution after updating from the primary sensor.

7.2 Incorporation of classification data

7.2.1 Problem formulation and solution

The problem here is the same as the baseline case except that with every measurement an imperfect classification feature $\underline{d}$ is available. Thus at some time step k, it is assumed that a set of data (Z, D) is received, where:

$$Z = \left\{ \underline{z}_j : j = 1, m \right\}$$

and

$$D \doteq \left\{ \underline{d}_j : j = 1, m \right\} .$$

(For convenience we shall omit the subscript k throughout this chapter.) Each classification feature $\underline{d}_j$ is independent of the values of $\underline{x}$ and Z, although it is known to correspond to measurement j. The value of $\underline{d}_j$ depends only on whether measurement j is true or false. It is assumed that the pdf of $\underline{d}_j$ conditional on measurement j being true is known, and it is denoted $p(\underline{d}_j|T)$. Similarly the pdf of $\underline{d}_j$ conditional on measurement j being false is known, and it is denoted $p(\underline{d}_j|F)$. With this knowledge, it is clear that the data set D may provide useful information as to which, if any, of the m measurements is the true one. We shall now derive the Bayesian filter which makes use of the classification features.

Following the reasoning of section 2.3.2, the posterior pdf of $\underline{x}$ after incorporation of the latest sensor data (Z, D), may be written:

$$p(\underline{x}|Z,D) = \sum_{i=1}^{n} \sum_{j=0}^{m} p\left(\underline{x}|\mathscr{H}'_{ij},Z,D\right) \Pr\left\{\mathscr{H}'_{ij}|Z,D\right\} . \qquad (7.1)$$

As in section 2.3.2, the explicit dependency on past data $\mathscr{D}$ has been omitted. First consider the pdf of $\underline{x}$ conditional on $\mathscr{H}'_{ij}$. Since the truth or falsehood of each measurement is specified by $\mathscr{H}'_{ij}$, the classification data does not contribute any extra information (it is independent of $\underline{x}$), so:

$$p\left(\underline{x} \mid \mathscr{H}'_{ij}, Z, D\right) = p\left(\underline{x} \mid \mathscr{H}'_{ij}, Z\right) ,$$

which is given as usual by equation (2.8). Thus the classification data only affects the weighting probabilities of the mixture distribution. By direct analogy with equation (2.10):

$$\Pr\left\{\mathscr{H}'_{ij} \mid Z, D\right\} = \frac{p\left(Z, D \mid \mathscr{H}'_{ij}\right) \Pr\left\{\Psi_j \mid \mathscr{H}_i\right\} \Pr\left\{\mathscr{H}_i\right\}}{p(Z, D)} , \qquad (7.2)$$

where $\mathscr{H}_i$ is a prior hypothesis and Ψ_j is the hypothesis that measurement j is true. Now since D is independent of Z, and D depends only on Ψ_j ,

$$p\left(Z, D \mid \mathscr{H}'_{ij}\right) = p\left(Z \mid \mathscr{H}'_{ij}\right) p\left(D \mid \Psi_j\right) .$$

Hence, comparing with equation (2.10) it can be seen that:

$$\Pr\left\{\mathscr{H}'_{ij} \mid Z, D\right\} \propto p\left(D \mid \Psi_j\right) \Pr\left\{\mathscr{H}'_{ij} \mid Z\right\} , \qquad (7.3)$$

where $\Pr\left\{\mathscr{H}'_{ij} \mid Z\right\}$ is the usual probability weighting for the baseline problem given by equation (2.18), and:

$$p\left(D \mid \Psi_j\right) = \begin{cases} p\left(\underline{d}_j \mid T\right) \prod\limits_{\substack{\ell=1 \\ \ell \neq j}}^{m} p\left(\underline{d}_\ell \mid F\right) & \text{for} \quad j \neq 0 \\[2ex] \prod\limits_{\ell=1}^{m} p\left(\underline{d}_\ell \mid F\right) & \text{for} \quad j = 0 , \end{cases} \qquad (7.4)$$

since the elements of D are independent. Thus dividing through by $p(D|\Psi_0)$, from (7.3) we obtain:

$$\Pr\left\{\mathscr{H}'_{ij}|Z,D\right\} = \begin{cases} L\left(\underline{d}_j\right) \Pr\left\{\mathscr{H}'_{ij}|Z\right\}/E & \text{if} \quad j \neq 0 \\ \Pr\left\{\mathscr{H}'_{i0}|Z\right\}/E & \text{if} \quad j = 0 \end{cases} \qquad \ldots\ldots (7.5)$$

where $L(\underline{d}) = p(\underline{d}|T)/p(\underline{d}|F)$ is a likelihood ratio and:

$$E = \sum_{i=1}^{n} \left[\Pr\left\{\mathscr{H}'_{i0}|Z\right\} + \sum_{\ell=1}^{m} L\left(\underline{d}_\ell\right) \Pr\left\{\mathscr{H}'_{ij}|Z\right\}\right] .$$

From equation (7.5) it is clear how the classification data may modify the original probability weightings of the baseline problem through the likelihood ratio $L(\underline{d})$. As usual, an estimate of $\underline{x}$ may be obtained from equation (7.1), and prediction forwards to obtain the prior pdf at the following time step follows from the state propagation equation as indicated in section 2.3.3.

A sub-optimal version of the filter described above may be implemented using the coarse acceptance test and one of the mixture reduction techniques of Chapter 3. The filter was first reported by Nagarajan *et al*[35] in 1984 and was implemented using the PDAF approximation. Note that minimal extra computation over that required for the baseline problem is necessary to incorporate the classification information.

The classification feature may also be of the discrete (0, 1) type, such that:

$$\Pr\{d|T\} = \begin{cases} P_T & \text{if} \quad d = 1 \\ 1 - P_T & \text{if} \quad d = 0 , \end{cases}$$

and

$$\Pr\{d|F\} = \begin{cases} P_F & \text{if} \quad d = 0 \\ 1 - P_F & \text{if} \quad d = 1 . \end{cases}$$

Thus $d = 1$ indicates that the measurement is likely to be true and the probability P_T of correctly recognizing a true measurement is known. Similarly $d = 0$ indicates that the measurement is likely to be false and the probability P_F of recognizing a false measurement is known. For this discrete case, the likelihood ratio in equation (7.5) should be replaced by:

$$L(d) = \frac{\Pr\{d|T\}}{\Pr\{d|F\}} = \left(\frac{P_T}{1 - P_F}\right)^d \left(\frac{1 - P_T}{P_F}\right)^{1-d} . \tag{7.6}$$

Note that if $P_T = P_F = \frac{1}{2}$, then $L(d) \equiv 1$ and in this case, as expected, the classification feature is ignored and the posterior probabilities are unaltered. If:

$$P_T = 1$$

and

$$0 < P_F < 1 ,$$

then

$$L(0) = 0$$

and

$$L(1) = 1/\left(1 - P_F\right) .$$

In this case the classifier always recognizes a true measurement but sometimes mistakes false for true. So any hypothesis for which $d_j = 0$ $(j \neq 0)$ is given a zero probability weighting via the likelihood ratio. If $P_F = 1$ and $0 < P_T < 1$, then the classifier always recognizes a false measurement but sometimes mistakes true for false. In this case the likelihood ratio defined by equation (7.6) is not defined when $d = 1$ and so it is not valid to divide through by $\Pr\left\{D|\Psi_0\right\}$ in equation (7.3) if any element of D is unity. However each probability weighting $\Pr\left\{\mathscr{H}'_{ij}|Z,D\right\}$ contains the factor:

$$\Pr\left\{D|\Psi_j\right\} = \prod_{\ell=1} \Pr\left\{d_\ell|\Psi_j\right\} ,$$

and if there exists an element of D such that $d_\ell = 1$, then $\Pr\left\{D|\Psi_j\right\}$ is non-zero only for $j = \ell$. Thus the true measurement is identified. Since the classifier always recognizes false measurements and there is at most one true measurement, only one element of D can be unity. However if P_T is less than one, the true measurement may not be recognized, so that all elements of D may be zero. In this case $\Pr\left\{D|\Psi_j\right\}$ is constant for $j \neq 0$. Note that if $P_F = 1$, the classifier will pick out the correct hypothesis Ψ_j on $100\ P_T$ % of occasions when the true measurement is present. Thus in a high density

of false measurements, a perfect false measurement discriminator may well be more useful than a perfect true measurement discriminator. If $P_F = P_T = 1$, the correct hypothesis is always identified.

7.2.2 Simulation

To demonstrate the possible improvement in tracking performance when classification data is available, the baseline example of section 4.2 has been extended to include a discrete (0, 1) type discriminator. The problem parameters used in Chapter 4 are retained:

$$\frac{q\Delta t^4}{r} = 1 \ , \ \rho r = 0.012 \ , \ P_D = 1 \ ,$$

and the performance of the classifier is defined by P_T and P_F as indicated above. Mixture reduction is carried out using either the PDAF or the Clustering Algorithm with $N_T = 20$.

Fig 7.1 shows the track survival time N_{AVE} and the average number of mixture components generated as a function of P_T when $P_F = P_T$. Also the average computation time per step is recorded in Table 7.1. As expected, N_{AVE} increases with the probability of correct classification and useful performance improvement may be obtained even with a mediocre discriminator. For example with $P_T = P_F = 0.7$, track lifetime of the CAF is increased by a factor of 2.5, although for the PDAF substantial improvement is not obtained until $P_T = P_F = 0.8$, when the improvement factor for both filters is about 3.4. Also the average processing time per step (Table 7.1) and the number of mixture components generated decrease as the performance of the discriminator improves. This is because the discriminator tends to suppress incorrect

hypothesis, and this helps to keep the acceptance region small so that fewer components are generated.

Fig 7.2 shows the effect of varying P_T with P_F fixed at 0.99, and Fig 7.3 shows results for varying P_F with $P_T = 0.99$. The corresponding results for the CAF are similar in these two cases, although the PDAF performs significantly better for small values of P_T with $P_F = 0.99$ than for small values of P_F with $P_T = 0.99$ (see previous section).

In each of Figs 7.1 to 7.3, the CAF track lifetime is always several times longer than that of the PDAF. However as P_F and P_T increase, the difference in performance between the two filters decreases (*cf* section 5.2).

7.3 Multiple sensors without classification data

7.3.1 Problem statement

In this section the baseline problem is extended to multiple sensors. Each of these has similar characteristics to the sensor described in Chapter 2 and no classification data is available. It is assumed that there are N_S independent sensors and that at each time step k , each sensor u produces m_u measurements:

$$Z_u = \left\{ \underline{z}_{uj} : j = 1 , m_u \right\} .$$

For each sensor u :

(i) At most one true measurement is produced with probability P_{Du} . This true measurement is an indpendent sample from the Gaussian pdf $\phi(\underline{z};\, H_u\underline{x},\, \Gamma_u)$.

(ii) False measurements are uniformly distributed over the surveillance region of the sensor. The density of false measurements is ρ_u .

Since each sensor is independent, data from each sensor may be incorporated sequentially using the update relations of section 2.3.2. This is convenient since the computer code for the single sensor problem may be employed with only minor modifications. This recursive solution is quite straightforward but for completeness it is included in Appendix E.

In any implementation of the filter it is necessary to control the proliferation of hypotheses. Depending on the density of false measurements, it may be feasible to apply a mixture reduction algorithm only once per time step, after measurements from all sensors have been processed (Fig 7.4b). In this case the order in which sensors are processed is irrelevant. Alternatively it may be desirable to carry out reduction after processing measurements from each sensor (Fig 7.4a). In this case the order in which sensors are processed may affect the performance of the filter. In the following section, these points are investigated by simulation for a two sensor filter.

7.3.2 Simulation example: a two sensor filter

To demonstrate the performance benefits that may be obtained with multiple sensors, the operation of a two sensor filter has been simulated for the tracking problem of section 4.2. The first sensor has parameters:

$$r_1 = q\Delta t^4 , \rho_1 = \frac{0.012}{q\Delta t^4} , P_{D1} = 1 ,$$

so that without sensor 2, the tracking problem would be identical to the example of Chapter 4. The second sensor is of the same type but may have different values for the parameters r_2, ρ_2 and P_{D2} . To facilitate comparison with the single sensor filter, the track loss criteria are identical to those given in section 4.3 and are based solely on sensor 1. Thus track loss through rejection of true measurements is only tested for sensor 1 and the tracking error reference is derived from the equivalent Kalman filter based on sensor 1 only (with $\rho_1 = 0$).

As indicated above, data sets from each sensor are incorporated sequentially. The two schemes for mixture reduction shown in Fig 7.4 have both been investigated using the Clustering Algorithm (CA) with the usual thresholds and $N_T = 20$. Also performance with the PDAF approximation has been studied. The PDAF must be applied directly after processing each sensor as retention of more than one component is not possible with this algorithm. This technique for incorporating multiple sensors using the PDAF has been implemented by Houlès and Bar-Shalom[14].

In the tracking simulation the parameters of sensor 2 were nominally chosen to have the same values as those of sensor 1 and then each of the parameters r_2, ρ_2 and P_{D2} were varied in turn. For reduction via the Clustering Algorithm, the average track survival times N_{AVE} (for 100 replications) with 95% confidence limits are shown in Figs 7.5 to 7.7. For each set of parameters, N_{AVE} is shown for the two sensor filter with reduction after processing *both* sensors (labelled TB) and with reduction after processing *each* sensor (labelled T12 when sensor 1 is processed first and labelled T21 when sensor 2 is processed first). Also results for the single sensor

filter using only sensor 1 (labelled S1) and using only sensor 2 (labelled S2) are shown for comparison. To maintain consistency, the tracking error reference of the track loss criterion for S2 is derived from the equivalent Kalman filter based on sensor 1 (with $\rho_1 = 0$). The average number of mixture components before and after application of the Clustering Algorithm is also shown in the figures. When the Clustering Algoithm is applied twice on each time step (for T12 and T21), both of these applications are included in the averages. Figs 7.8 to 7.10 show similar results for the PDAF approximation, except TB using the Clustering Algorithm has been included for comparison. Average cpu time per time step is given in Table 7.2 for all of these results.

In many cases, employing two sensors gives an increase in track lifetime N_{AVE} with respect to a filter using measurements from only one of the two sensors. The greatest improvement factor is obtained when the two sensors are identical, for which N_{AVE} from TB exceeds N_{AVE} from the single sensor CAF by a factor of about 8.5. When there is a large discrepancy between the quality of the two sensors, the performance of the two sensor filters does not usually differ significantly from the best of the single sensor filters. However in two cases where the track lifetime of S2 is greater than 15 times that of S1 ($\rho_2 = 0.001/q\Delta t^4$ and $\rho_2 = 0.002/q\Delta t^4$ in Figs 7.6 and 7.9), the two sensor filter T12 is outperformed by S2 for both the CAF and the PDAF. In each of these two cases, TB using the Clustering Algorithm is still better than the CAF using sensor 2 alone. Thus it appears that when the quality of the two sensors is very dissimilar, it is important to retain the detailed structure of the mixture between processing data from the sensors. Presumably this allows the good

sensor to selectively reinforce or suppress components generated by the poor sensor.

In all cases track lifetime for TB is greater than or not significantly different from N_{AVE} for T12 or T21, for the Clustering Algorithm. Also TB which uses the Clustering Algorithm, always gives a track lifetime at least five times longer than that of T12 or T21 using the PDAF. For both the CAF and the PDAF, N_{AVE} from T12 is usually similar to N_{AVE} from T21. When a significant difference does occur, the longer track survival time is usually obtained when the better sensor is processed first. The one exception to this is for the CAF with $r_2 = 0.01\ q\Delta t^4$ (Fig 7.5).

For the Clustering Algorithm, the average cpu time per step for the two sensor filters is almost always much less than that of the CAF employing only the poor sensor, and greater than the CAF employing only the good sensor (see Table 7.2). For identical sensors, the cpu time per step for TB is 16% greater than that of the single sensor CAF, while T12 and T21 give a 25% saving in cpu time. These computation times are closely related to the average number of mixture components generated by the filters. The effect on cpu time of incorporating a second sensor is broadly similar for the PDAF approximation, except that for $P_{D2} < 1$, the two sensor filters are slower than the PDAF using sensor 2 alone (which performs very poorly).

It should be remembered that the above observations only apply to the example simulated here. However it is quite likely that the broad conclusions apply to a wide range of examples. Detailed results, such as the percentage of cpu time saved by employing two identical sensors rather than one of them, are likely to be problem dependent.

7.4 Incorporation of data from an auxiliary sensor with a classification capability

7.4.1 Problem statement

This data fusion example has been chosen to show how data from a secondary or auxiliary sensor may be used to assist a primary sensor with modest changes to the tracking filter. This example is an extension of the sector surveillance problem of Chapter 6, and as already described it is assumed that the primary sensor produces measurements in polar co-ordinates. The auxiliary sensor produces bearing only measurements, but a classification flag is associated with each of these. Since the auxiliary sensor does not supply range, on its own it would give poor tracking performance. The auxiliary sensor is co-located with the primary sensor at the origin and measurement sets are produced coincidently by both sensors. It can be seen that this problem includes elements from each of the previous sections.

The auxiliary sensor produces false measurements which are uniformly distributed in bearing over the surveillance sector with a density ρ_2 per radian. A true measurement has a Gaussian distribution about the actual target bearing with a standard deviation of σ_2 radians, and the probability of detecting the target is P_{D2}. The classification flag associated with each measurement is of the discrete (0, 1) type. A value of one indicates that the measurement has been classified true, while zero indicates that it has been classified false. The probability of correctly recognizing a true measurement is P_T, and P_F is the probability of correctly recognizing a false measurement. As in section 7.2, the classification flag is independent of the value of the measurement.

7.4.2 A sub-optimal filter

The main idea behind the design of this filter is to use data from the auxiliary sensor to modify only the probability weights of the mixture distribution resulting from the primary sensor measurements. So the auxiliary sensor data is to be used either to reinforce or to weaken the weightings of the mixture components. The mixture components themselves are not changed. This approach avoids the usual splitting of components when measurements from the second sensor are incorporated.

After processing the measurements Z_1 from the primary sensor at some time step, the posterior pdf of $\underline{x}$ is given by (following section 2.3.2):

$$p\left(\underline{x} | Z_1\right) = \sum_{i=1}^{n} \sum_{j=0}^{m_1} p\left(\underline{x} | \psi_{1j}, \mathscr{H}_i, Z_1\right) \Pr\left(\psi_{1j}, \mathscr{H}_i | Z_1\right) , \qquad \ldots\ldots (7.7)$$

where ψ_{1j} is a hypothesis on the measurements from the primary sensor (see Appendix E). Since the sensor measurements are in polar co-ordinates, the extended Kalman filter approximation is used to evaluate the components and [illegible]bility weights of this Gaussian mixture (see section 6.1). The data from the auxiliary sensor is denoted:

$$\left(Z_2, D_2\right) ,$$

where D_2 is the set of (0, 1) type classification features:

$$D_2 = \left\{ d_{2\ell} : \ell = 1, m_2 \right\} .$$

After incorporating data from the auxiliary sensor, the posterior pdf of $\underline{x}$ becomes:

$$p\left(\underline{x} | Z_1, Z_2, D_2\right) = \sum_{i=1}^{n} \sum_{j=0}^{m_1} \sum_{\ell=0}^{m_2} p\left(\underline{x} | \psi_{1j}, \psi_{2\ell}, \mathscr{H}_i, Z_1, Z_2\right)$$

$$\times \Pr\left\{\psi_{1j}, \psi_{2\ell}, \mathscr{H}_i | Z_1, Z_2, D_2\right\} , \quad (7.8)$$

where $\psi_{2\ell}$ is a hypothesis on the measurements from the auxiliary sensor. (Note that D_2 is not required in first term on the RHS of equation (7.8) - see section 7.2.1.) We now impose the simplifying assumption that the effect of the auxiliary sensor measurements Z_2 on the components of the mixture (7.7) can be ignored, *ie*:

$$p\left(\underline{x} | \psi_{1j}, \psi_{2\ell}, \mathscr{H}_i, Z_1, Z_2\right) \simeq p\left(\underline{x} | \psi_{1j}, \mathscr{H}_i, Z_1\right) .$$

In this case equation (7.8) may be written:

$$p\left(\underline{x} | Z_1, Z_2, D_2\right) = \sum_{i=1}^{n} \sum_{j=0}^{m_1} p\left(\underline{x} | \psi_{1j}, \mathscr{H}_i, Z_1\right) \Pr\left\{\psi_{1j}, \mathscr{H}_i | Z_1, Z_2, D_2\right\} ,$$

$$\ldots\ldots \quad (7.9)$$

where $\Pr\left\{\psi_{1j},\mathscr{H}_i|Z_1,Z_2,D_2\right\} = \sum_{\ell=0}^{m_2} \Pr\left\{\psi_{1j},\psi_{2\ell},\mathscr{H}_i|Z_1,Z_2,D_2\right\}$.

After applying Bayes theorem and deleting redundant dependencies it can be shown that:

$$\Pr\left\{\psi_{1j},\mathscr{H}_i|Z_1,Z_2,D_2\right\} = \Pr\left\{\psi_{1j},\mathscr{H}_i|Z_1\right\} \sum_{\ell=0}^{m_2} F_{ij\ell} \quad , \qquad (7.10)$$

where $$F_{ij\ell} = \frac{\Pr\left\{D_2|\psi_{2\ell}\right\}p\left(Z_2|\psi_{1j},\psi_{2\ell},\mathscr{H}_i,Z_1\right)\Pr\left\{\psi_{2\ell}|\psi_{1j},\mathscr{H}_i,Z_1\right\}}{E} \quad , \qquad \ldots\ldots (7.11)$$

and E is the normalizing denominator chosen so that the summation of the RHS of equation (7.10) over i and j is unity. (Note that D_2 is independent of the past, so $\Pr\left\{D_2|\psi_{2\ell}\right\}$ does not include a dependency on $\mathscr{H}_i$.) Thus the resulting filter is the same as the usual single sensor filter except that each probability weighting is modified by the factor:

$$\sum_{\ell=0}^{m_2} F_{ij\ell} \quad .$$

For the problem of section 7.4.1, the auxiliary sensor data consists of bearing measurements, each with an associated classification

flag. Thus using an amalgam of results from sections 2.3.2, 6.1, 7.2.1 and Appendix E it can be shown that equation (7.11) is given by:

$$F_{ij\ell} = \begin{cases} \dfrac{\left(\dfrac{P_T}{1 - P_F}\right)^{d_{2\ell}} \left(\dfrac{1 - P_T}{P_F}\right)^{1-d_{2\ell}}}{E'} \dfrac{\mathcal{N}\left(\theta_{2\ell};\bar{\theta}_{ij},\sigma^2_{ij}\right)}{} & \text{if} \quad \ell \neq 0 \\ \dfrac{\left(1 - P_{D2}\right) \rho_2}{P_{D2}\ E'} & \text{if} \quad \ell = 0 \end{cases} \qquad \text{...... (7.12)}$$

where E' is the normalizing denominator, chosen so that:

$$\sum_{i=1}^{n} \sum_{j=0}^{m_1} \Pr\left\{\psi_{1j}, \mathcal{H}_i | Z_1\right\} \sum_{\ell=0}^{m_2} F_{ij\ell} = 1 \quad .$$

Also $\theta_{2\ell}$ is the auxiliary bearing measurement ℓ. $\bar{\theta}_{ij}$ is the expected value of the true auxiliary measurement under hypothesis $(\psi_{1j}, \mathcal{H}_i)$, and it is given by:

$$\bar{\theta}_{ij} = \tan^{-1}\left(\hat{y}_{ij}/\hat{x}_{ij}\right)$$

where $(\hat{x}_{ij}, \hat{y}_{ij})$ is the mean target position of the mixture component of equation (7.7) corresponding to hypothesis $(\psi_{1j}, \mathcal{H}_i)$. σ^2_{ij} is the variance of the innovation $(\theta_{2\ell} - \bar{\theta}_{ij})$ under hypothesis $(\psi_{1j}, \mathcal{H}_i)$, and from equations (6.5) and (6.6) it is given by:

$$\sigma_{ij}^2 = \frac{1}{\bar{r}_{ij}^2}\left(p_{11}\sin^2\bar{\theta}_{ij} + p_{33}\cos^2\bar{\theta}_{ij} - p_{13}\sin 2\bar{\theta}_{ij}\right) + \sigma_2^2$$

where $\bar{r}_{ij}^2 = \hat{x}_{ij}^2 + \hat{y}_{ij}^2$

and p_{11}, p_{33} and p_{13} are elements of the symmetric matrix P_{ij} which is the covariance of the mixture component of (7.7) corresponding to hypothesis $(\psi_{1j}, \mathscr{H}_i)$. If σ_{ij} is large, then there is likely to be a large uncertainty in the association of auxiliary measurement to mixture component, and the extra data is unlikely to be informative. However if σ_{ij} is small so that the Gaussian factor in equation (7.12) is selective, the auxiliary data may provide useful extra information.

From above it can be seen that certain elements of the mean $\hat{\underline{x}}_{ij}$ and the covariance P_{ij} of each component of equation (7.7) are required for the evaluation of $F_{ij\ell}$. These terms are already available for an implementation using the Clustering Algorithm, so that incorporating the auxiliary sensor data is a small computational overhead. However, for the standard PDAF, $\hat{\underline{x}}_{ij}$ and P_{ij} are not explicitly evaluated and so for this filter the extra computation requirement is significant. To reduce the processor load for the PDAF it is suggested that components with very low probability weights are discarded before calculating the modifying factor:

$$\sum_{\ell=0}^{m_2} F_{ij\ell} \ .$$

In the simulation of the following section, components with probability weights below 0.001 are ignored for the PDAF. Also for both the CAF and the PDAF, an acceptance test is applied to the auxiliary measurements for each component of equation (7.7). Mixture reduction is applied after modifying the probability weights, and prediction forwards to the next time step follows as usual from the state propagation equation.

7.4.3 Simulation

Simulation studies have been carried out to demonstrate the possible improvement in tracking performance through sub-optimal processing of auxiliary sensor measurements. The standard parameters of Table 6.1 have been assumed for the target trajectory and for the primary sensor, except that the density of false measurements for the primary sensor has been increased to $\rho = 30\ \text{km}^{-1}\ \text{rad}^{-1}$. The performance of the auxiliary sensor is described by five parameters:

(i) the standard deviation of the true measurement bearing error σ_2 (radians),

(ii) the density of false measurements ρ_2 (radians^{-1}),

(iii) the probability of correctly recognizing a true measurement P_T ,

(iv) the probability of correctly recognizing a false measurement P_F .

(v) the probability of detecting the target P_{D2} .

For this simulation we have set $P_T = P_F$, and the following standard set of parameters for the auxiliary sensor has been chosen:

$$\sigma_2 = 0.01745 \text{ rad}$$

$$\rho_2 = 45 \text{ rad}^{-1}$$

$$P_T = P_F = 0.9$$

$$P_{D2} = 1 \ .$$

Thus the standard deviation of the true measurement σ_2 is the same as σ_θ for the primary sensor, and ρ_2 is related to the density of primary sensor false measurements by:

$$\rho_2 = \frac{1}{12} \rho \left(r_2 - r_1 \right) \ ,$$

where $r_2 - r_1 = 18$ km is the range extent of the surveillance sector. Each of the parameters σ_2, ρ_2 and P_T has been varied in turn while keeping the parameters of the primary sensor fixed. Figs 7.11 to 7.13 show the percentage of tracks maintained by the Auxiliary Sensor filter out of 100 replications for each set of parameters tested. For the track maintenance criterion of (6.1), σ_x and σ_y are obtained from the equivalent Kalman filter based on the primary sensor only. The average processing time for a single step and the error statistic $\bar{E}$ (see section 6.3.1) are given in Table 7.3.

Figs 7.11 to 7.13 clearly show that the Auxiliary Sensor filter can give a significant performance improvement over the primary sensor alone. This is most apparent for the PDAF which can only retain 1% of the tracks without the auxiliary sensor. As would be expected, performance deteriorates with increasing σ_2 and ρ_2, so as these parameters become large, performance approaches the primary sensor alone case (Figs 7.11 and 7.12). Also filter performance improves as P_T increases (Fig 7.13). For the case $P_T = P_F = 0.5$, the classifier

supplies no useful information. However for $P_T = P_F = 1$, the true auxiliary measurement is always identified so that the presence of false auxiliary measurements is irrelevant (*cf* filter performance for $\rho_2 = 0.36\ \text{rad}^{-1}$ in Fig 7.12 for which false measurements are sparse).

Table 7.3 shows that if the performance of the auxiliary sensor is good (σ_2 or ρ_2 low, or P_T high), the incorporation of the extra data reduces the average computation time for the CAF: the extra information enables the filter to reduce the number of retained components (see Figs 7.11 to 7.13). Processing time is always greater for the Auxiliary Sensor PDAF than for the standard single sensor PDAF. This is because the mean and covariance of each mixture component must be explicity calculated for the Auxiliary Sensor PDAF implementation (see previous section). When the density ρ_2 of the auxiliary false measurements is large, the processing times for the Auxiliary Sensor filters are several times greater than those of the standar filters.

Examination of the error statistic $\bar{F}$ in Table 6.3 shows that for lost tracks, the filters significantly underestimate their tracking error (as is also the case for the standard filters, see section 6.3.1). For CAF held tracks, with the exception of the cases $\sigma_2 = 0.005$ rad and $\sigma_2 = 0.04$ rad , $\bar{E}$ is always within 50% of the 'correct' value of four. However for the PDAF, the values of $\bar{E}$ show a much greater spread about four, with a tendency for $\bar{E}$ to increase with the performance of the auxiliary sensor.

7.5 Conclusions

In this chapter we have shown how Bayesian filters may be applied to the data fusion problem. Incorporating data from an extra sensor or an imperfect measurement classifier may significantly improve tracking

performance and reduce processing time. However if the performance of the additional sensor is very inferior to the original sensor, a large processing overhead may result in only a minor performance improvement.

Table 7.1

PROCESSOR TIMINGS FOR FILTERS
WITH CLASSIFICATION FLAG

Classification parameters		Average cpu time for single step (ms)	
P_T	P_F	CAF	PDAF
0.50	0.50	5.930	1.120
0.60	0.60	5.420	1.140
0.70	0.70	3.710	0.659
0.80	0.80	2.210	0.460
0.90	0.90	1.070	0.206
0.95	0.95	0.675	0.199
0.99	0.99	0.519	0.195
0.30	0.99	3.240	0.379
0.50	0.99	1.920	0.236
0.90	0.99	0.679	0.199
0.95	0.99	0.594	0.196
0.99	0.30	3.240	0.813
0.99	0.50	2.010	0.6[illegible]3
0.99	0.90	0.677	0.198
0.99	0.95	0.581	0.197

Table 7.2

TWO SENSOR FILTERS: AVERAGE CPU TIME TO PERFORM FILTERING FOR A SINGLE TIME STEP (ms)

Parameters for sensor 2			Clustering Algorithm approximation					PDAF approximation			
			Single sensor		Both sensors			Single sensor		Both sensors, approximation after processing *each* sensor	
					Approximation after processing *each* sensor		Approx. after processing *both*				
P_{D2}	ρ_2	r_2	S1 only	S2 only	T12	T21	TB	S1 only	S2 only	T12	T21
1.0	1.2	10^{-4}	5.93	0.431	0.918	0.873	1.07	1.12	0.187	0.331	0.328
1.0	1.2	10^{-3}	"	0.643	1.530	1.400	2.09	"	0.414	0.357	0.336
1.0	1.2	10^{-2}	"	5.930	4.360	4.310	6.82	"	1.120	0.546	0.510
1.0	1.2	10^{-1}	"	149.000	10.900	13.400	22.50	"	1.530	1.450	1.690
1.0	1.2	1	"	42.700	16.050	40.300	141.00	"	1.480	2.540	2.010
.0	0.1	10^{-2}	"	0.340	2.170	1.510	2.83	"	0.160	0.345	0.317
1.0	0.2	10^{-2}	"	0.521	2.300	1.710	3.08	"	0.161	0.350	0.325
1.0	1.2	10^{-2}	"	5.930	4.360	4.310	6.82	"	1.120	0.546	0.510
1.0	3.0	10^{-2}	"	803.000	8.960	11.700	18.00	"	2.740	1.800	1.680
1.0	10.0	10^{-2}	"	1510.000	28.900	51.300	74.50	"	6.500	5.137	5.710

Table 7.2 (concluded)

Parameters for sensor 2			Clustering Algorithm approximation					PDAF approximation			
			Single sensor		Both sensors			Single sensor		Both sensors, approximation after processing *each* sensor	
					Approximation after processing *each* sensor		Approx. after processing *both*				
P_{D2}	ρ_2	r_2	S1 only	S2 only	T12	T21	TB	S1 only	S2 only	T12	T21
0.05	1.2	10^{-2}	5.93	17.2	9.29	9.83	14.90	1.12	1.64	2.170	2.200
0.2	1.2	10^{-2}	"	22.6	10.40	11.40	16.80	"	1.30	2.010	1.860
0.6	1.2	10^{-2}	"	25.0	8.60	10.50	13.70	"	1.10	1.200	1.230
1.0	1.2	10^{-2}	"	5.93	4.36	4.31	6.82	"	1.12	0.546	0.510

Table 7.3

PROCESSOR TIMINGS AND ERROR STATISTIC $\bar{E}$ FOR AUXILIARY SENSOR FILTER

Parameters of auxiliary sensor			Tracks held or lost	Average cpu time for single step (ms)		Error statistic $\bar{E}$	
σ_2 (rad)	ρ_2 (rad^{-1})	$P_T = P_F$		CAF	PDAF	CAF	PDAF
Primary sensor only			H L	15.00 52.80	1.13* 2.72	3.660 2081.000	3.142* 6.694
0.005	45.00	0.9	H L	9.11 9.33	2.93 3.66	22.930 2779.000	16.490 566.600
0.01	"	"	H L	10.00 9.57	2.94 3.95	4.262 2419.000	71.080 444.100
0.01745	"	"	H L	11.50 11.50	3.85 4.96	3.691 1713.000	27.560 258.400
0.04	"	"	H L	14.50 14.60	5.73 7.37	435.420 1742.000	5.581 88.180
0.07	"	"	H L	16.90 22.80	8.32 9.61	3.558 3069.000	4.346 128.300
0.1	"	"	H L	18.60 22.70	5.93 12.60	3.601 2784.000	4.204 69.230
0.2	"	"	H L	21.25 64.22	10.37* 19.22	3.643 2159.000	3.339* 71.820
0.01745	0.36	0.9	H L	8.54 7.72	1.95 2.34	6.106 1377.000	17.490 39.650
0.01745	1.80	"	H L	8.64 8.27	1.97 2.41	3.718 1344.000	18.240 199.300
0.01745	9.00	"	H L	9.20 9.52	2.23 2.67	3.707 6627.000	10.650 422.200
0.01745	45.00	"	H L	11.50 11.50	3.85 4.96	3.691 1713.000	27.560 258.400
0.01745	180.00	"	H L	19.00 20.90	10.30 18.60	3.590 2362.000	2.711 95.980
0.01745	720.00	"	H L	44.40 126.00	23.90 74.00	3.927 2522.000	2.678 96.390
0.01745	1440.00	"	H L	76.20 79.40	68.50 145.00	3.749 2495.000	21.140 155.400

Table 7.3 (concluded)

Parameters of auxilliary sensor			Tracks held or lost	Average cpu time for single step (ms)		Error statistic $\bar{E}$	
σ_2 (rad)	ρ_2 (rad^{-1})	$P_T = P_F$		CAF	PDAF	CAF	PDAF
0.01745	45.0	0.5	H	15.00	4.44	4.437	2.408
			L	21.60	9.19	9.188	41.770
"	"	0.6	H	16.70	4.41	3.585	2.383
			L	18.10	9.08	3012.000	23.730
"	"	0.7	H	14.20	5.69	3.607	3.677
			L	15.00	8.10	6104.000	41.760
"	"	0.8	H	12.90	4.09	3.585	2.761
			L	14.70	6.79	2896.000	190.900
"	"	0.9	H	11.50	3.85	3.691	27.560
			L	11.50	4.96	1713.000	258.400
"	"	0.95	H	10.70	3.17	3.745	31.050
			L	10.50	4.22	2183.000	201.500
"	"	0.99	H	9.87	2.89	3.711	16.210
			L	9.55	3.70	1622.000	204.000
"	"	1.0	H	9.48	2.76	3.836	20.030
			L	9.92	3.31	3709.000	388.900

* Indicates a small sample (less than five replications)

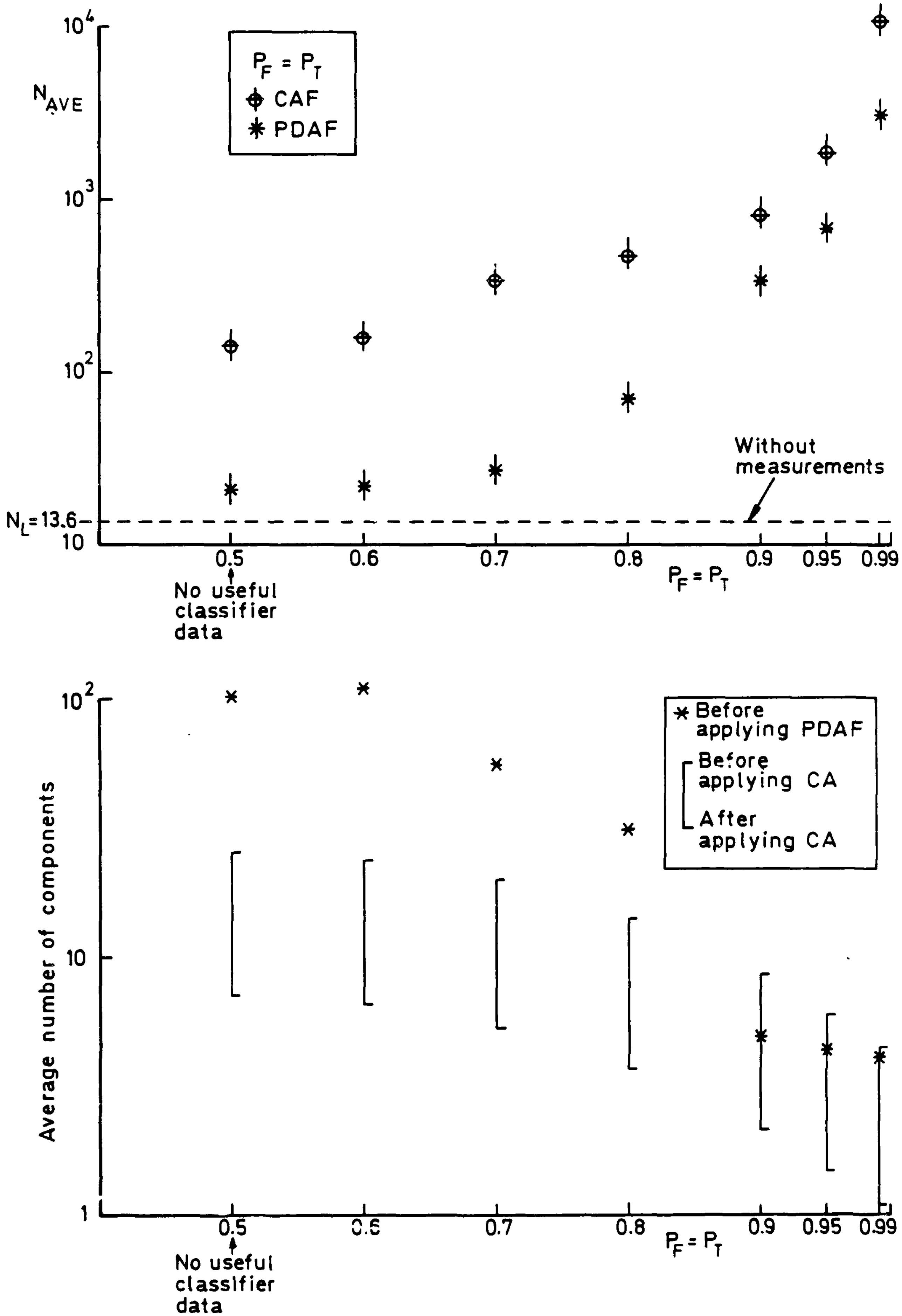


Fig 7.1 Filter performance with classification flag ($P_F = P_T$)

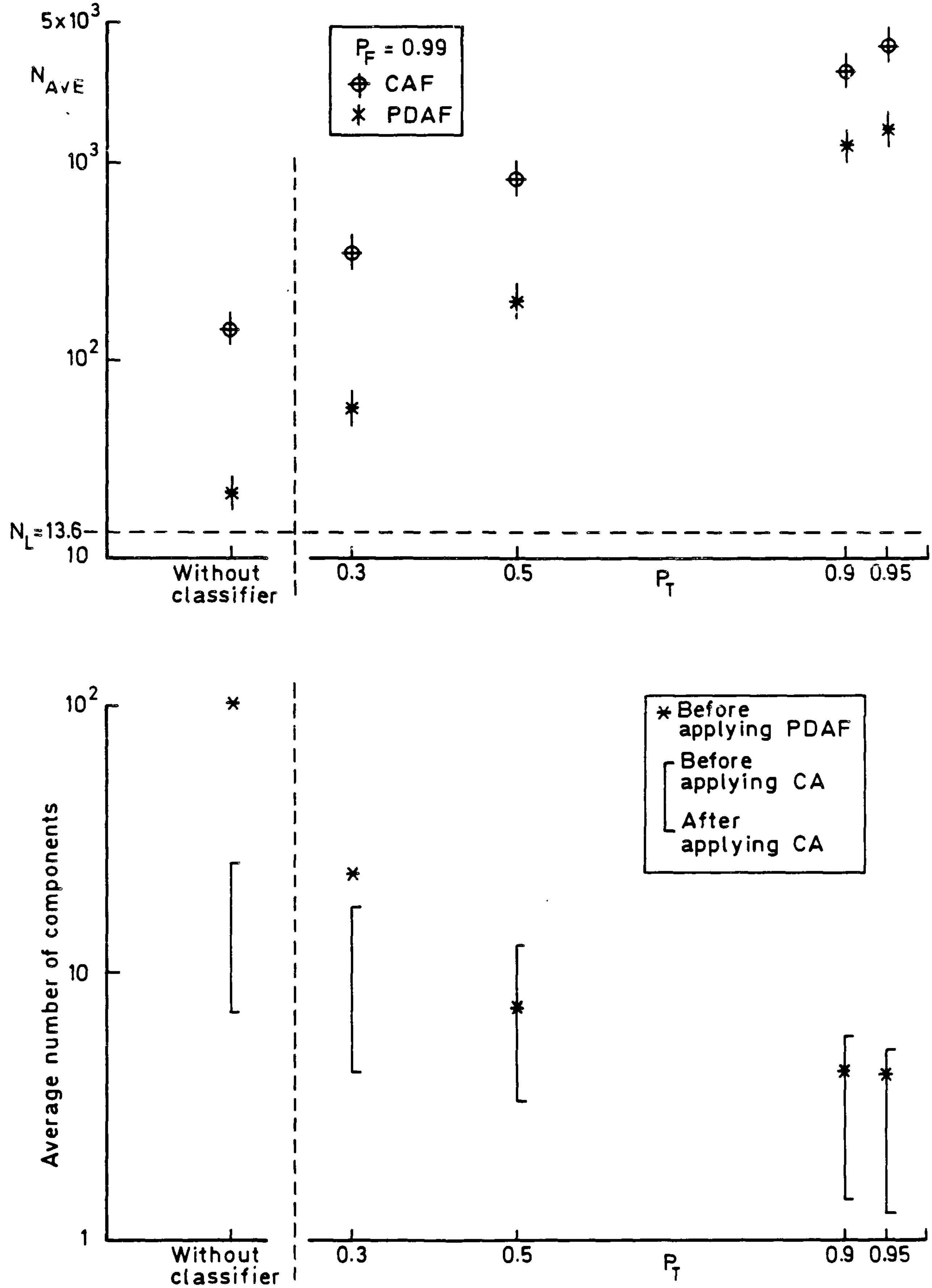


Fig 7.2 Filter performance with classification flag (P_F = 0.99)

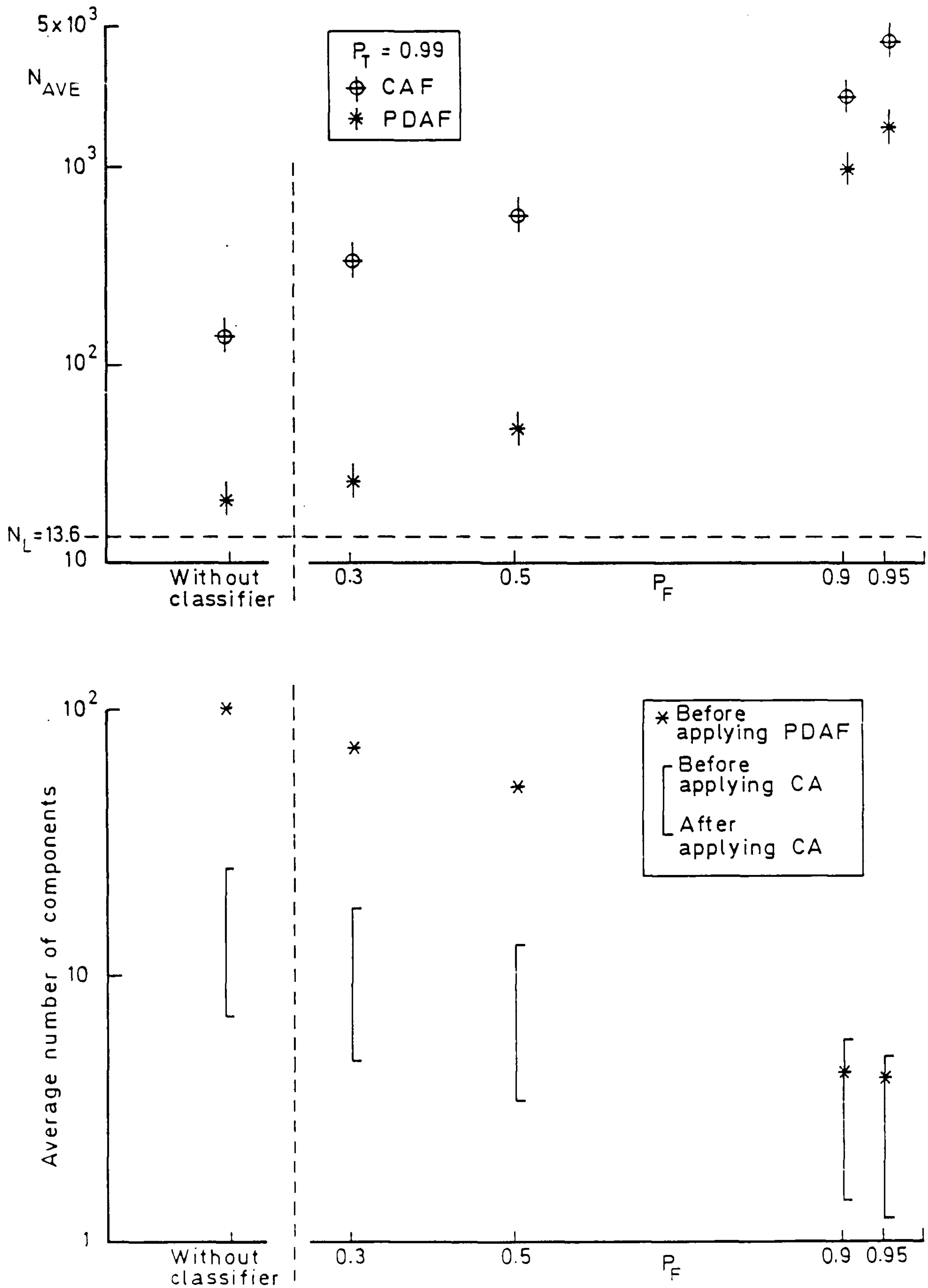
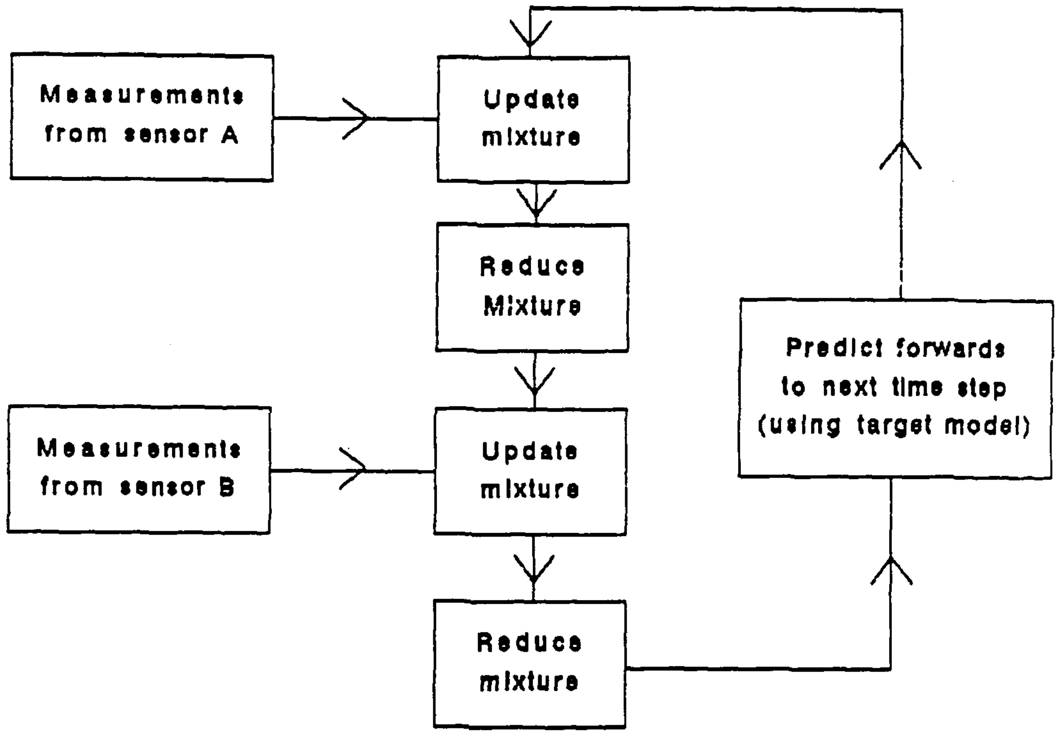
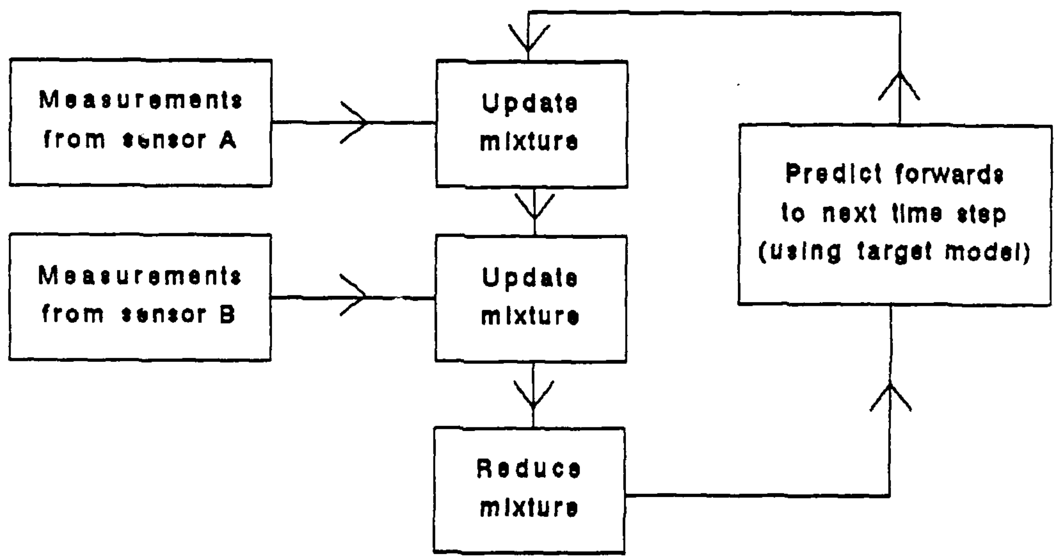


Fig 7.3 Filter performance with classification flag (P_T = 0.99)



a) Reduction applied after processing <u>each</u> sensor



b) Reduction applied after processing <u>both</u> sensors

Fig 7.4 Mixture reduction schemes for two sensor filter

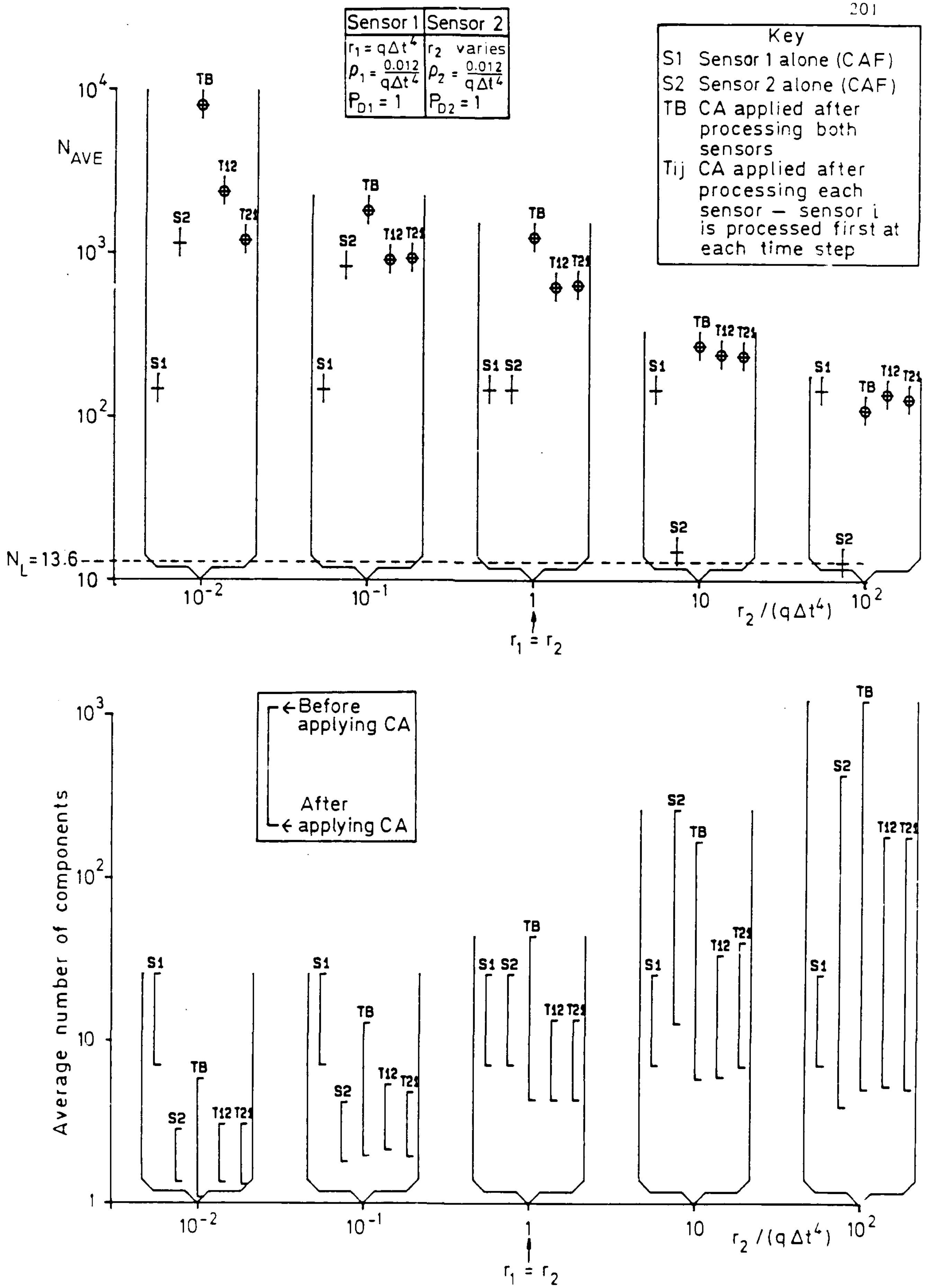


Fig 7.5 Two sensor CAF performance (ρ_2 and P_{D2} constant, r_2 varies)

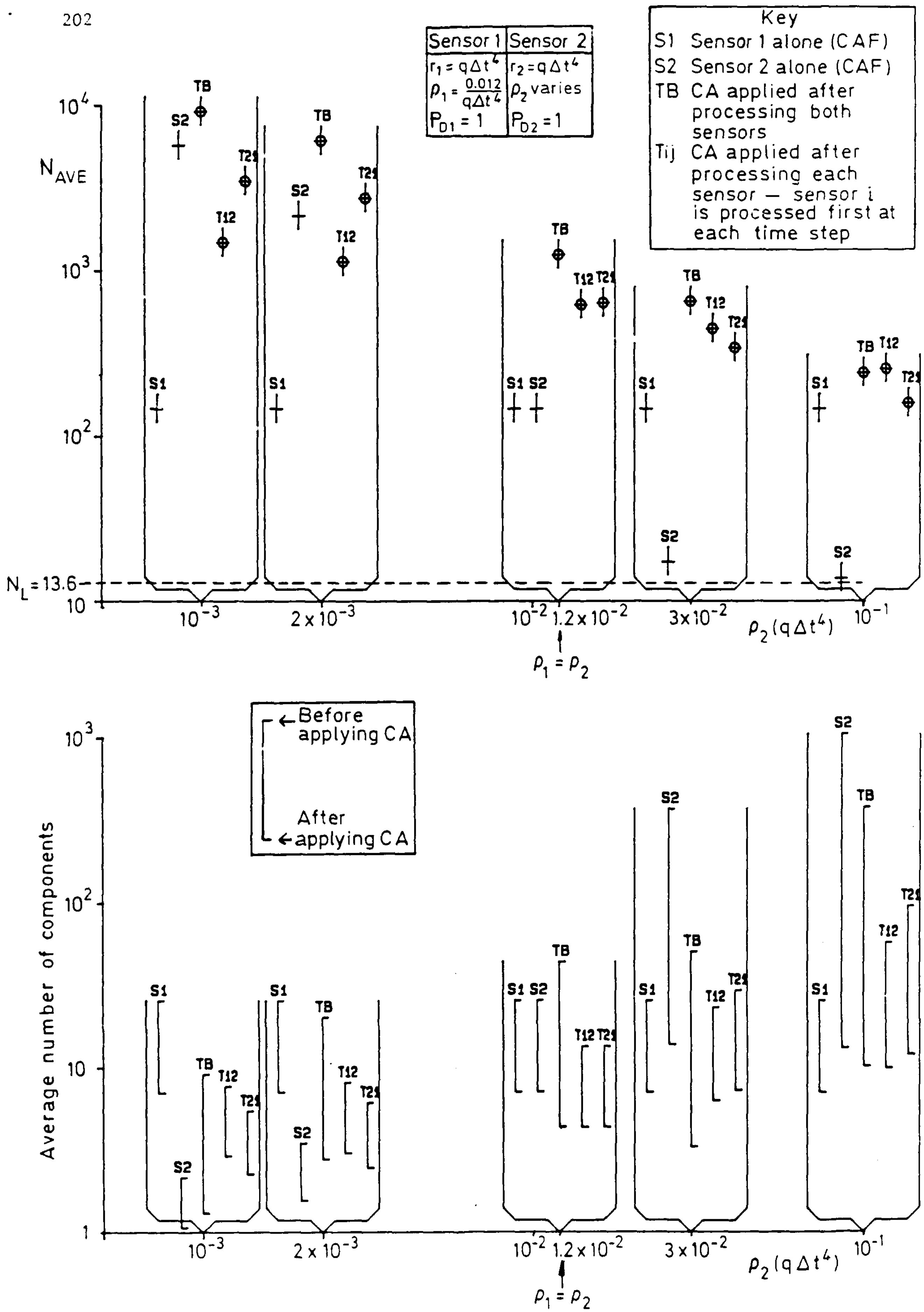


Fig 7.6 Two sensor CAF performance (r_2 and P_{D2} constant, ρ_2 varies)

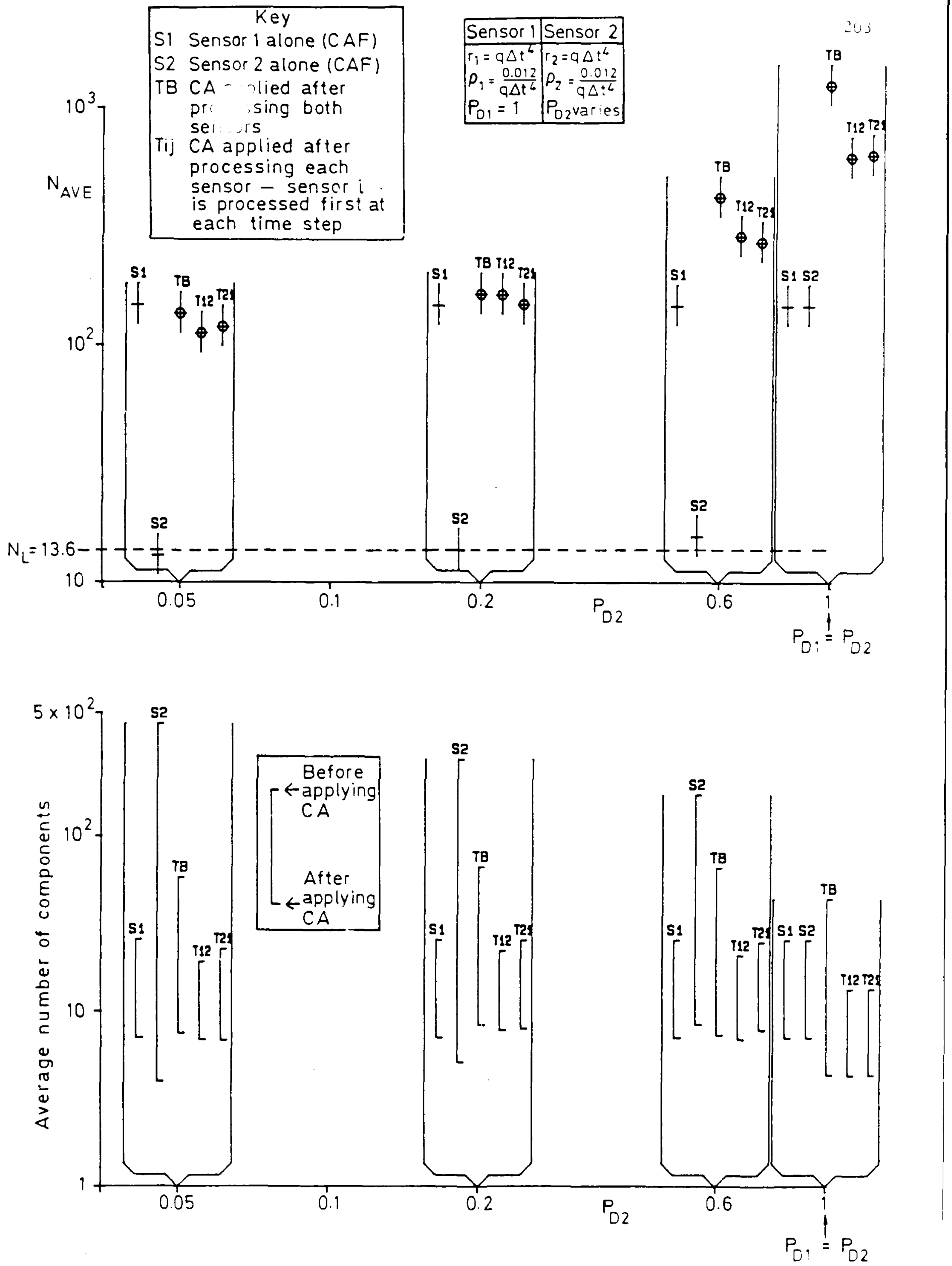


Fig 7.7 Two sensor CAF performance (r_2 and ρ_2 constant, P_{D2} varies)

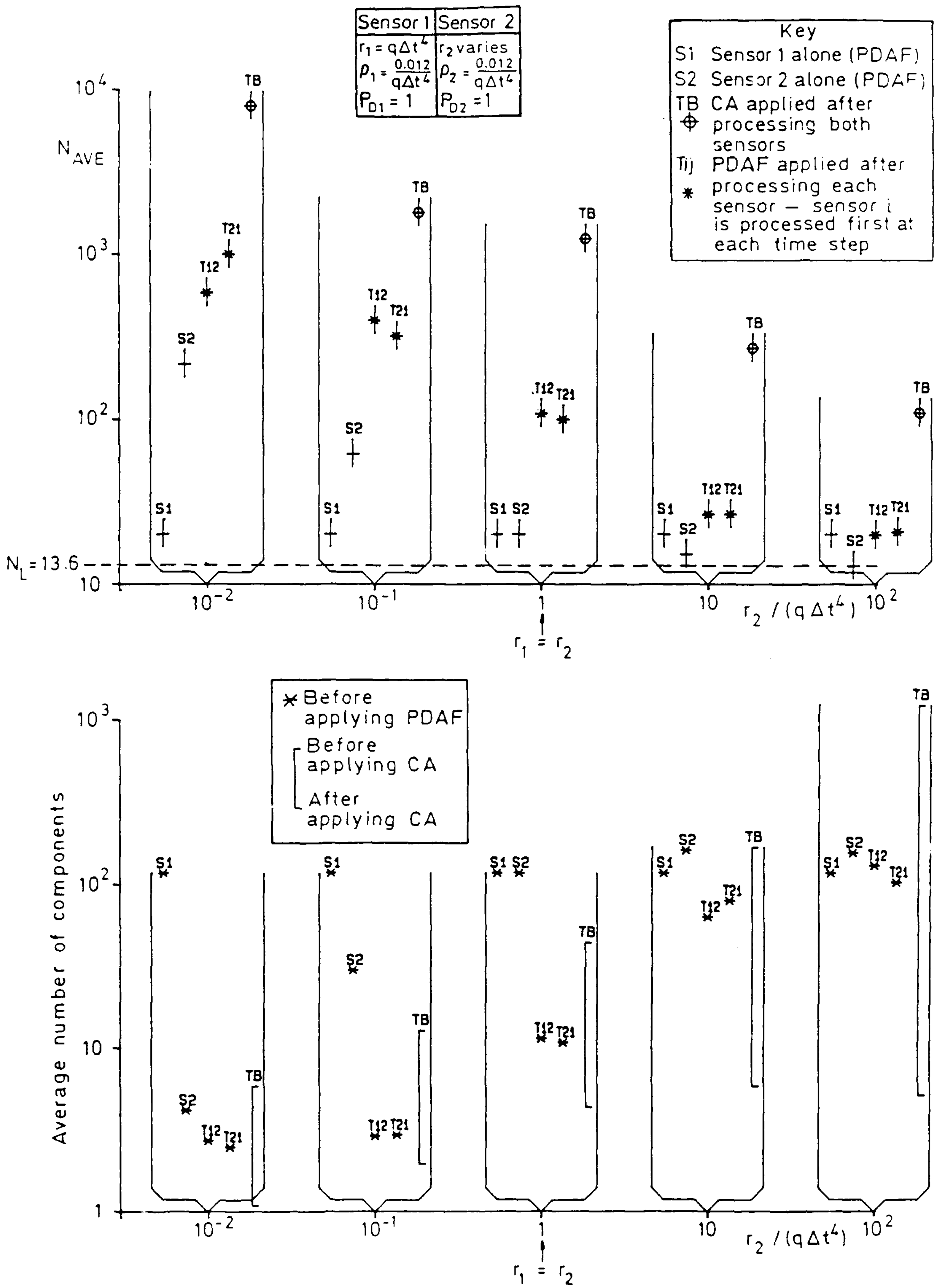


Fig 7.8 Two sensor PDAF performance (ρ_2 and P_{D2} constant, r_2 varies)

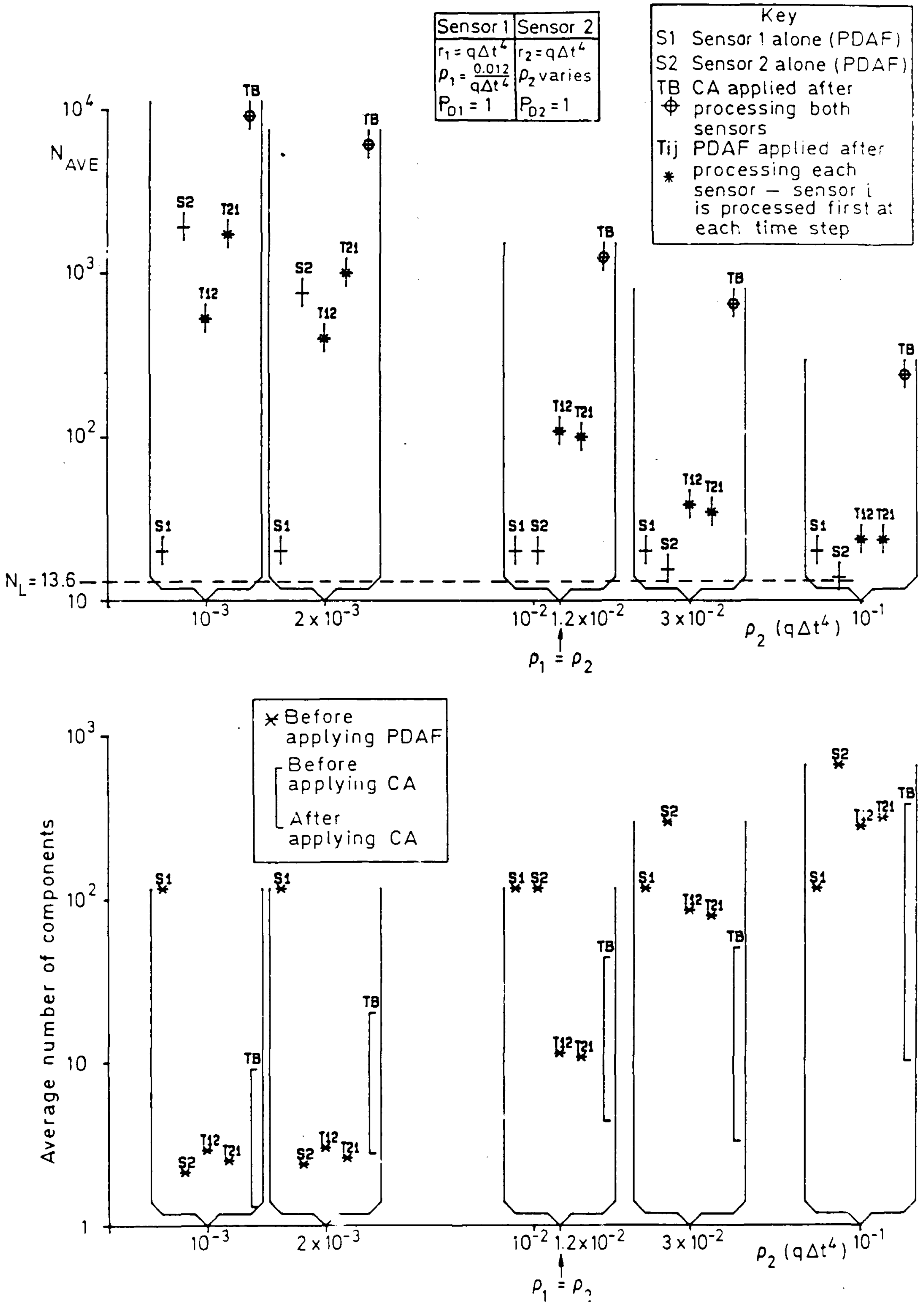


Fig 7.9 Two sensor PDAF performance (r_2 and P_{D2} constant, ρ_2 varies)

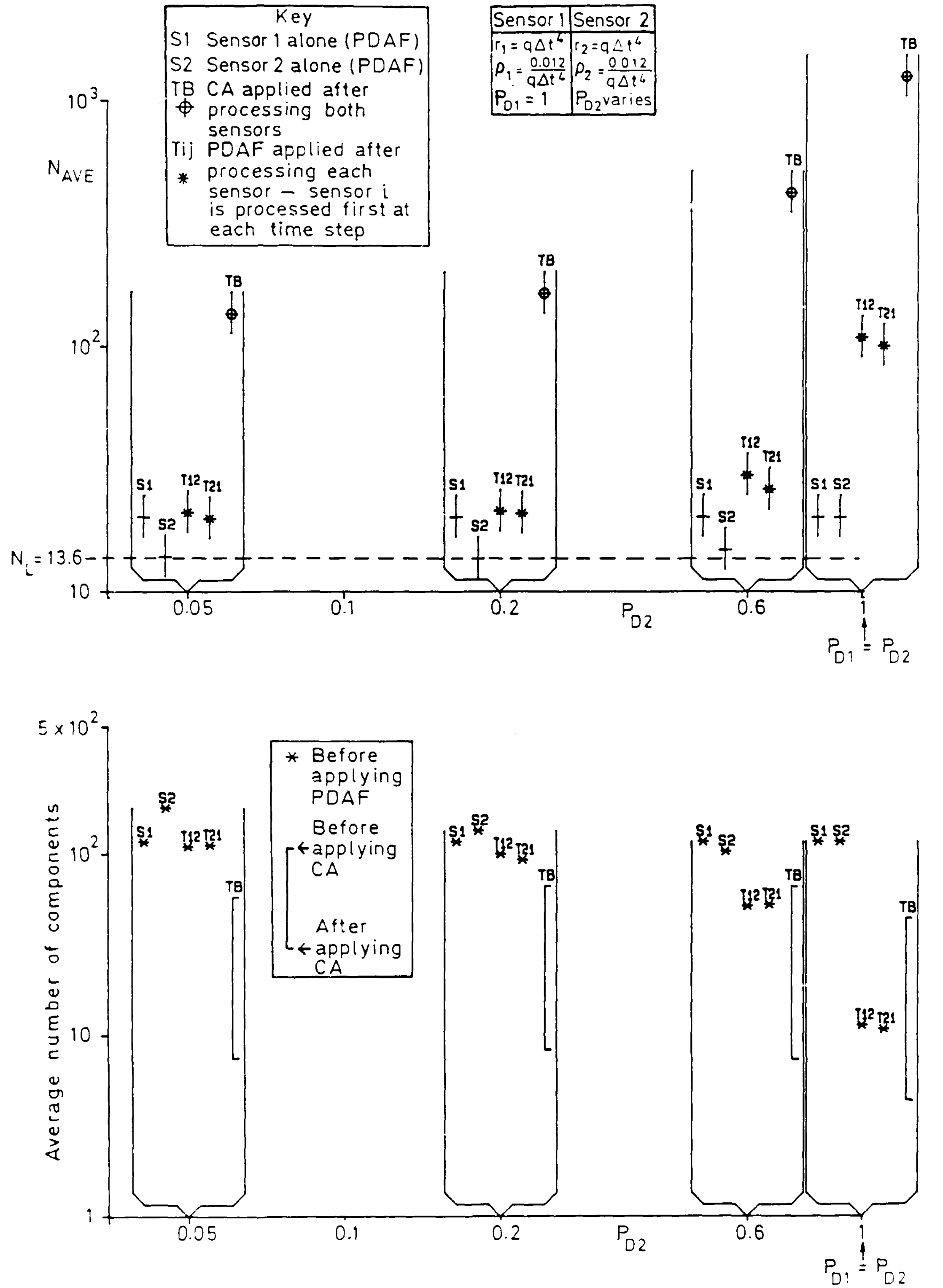


Fig 7.10 Two sensor PDAF performance (r_2 and ρ_2 constant, P_{D2} varies)

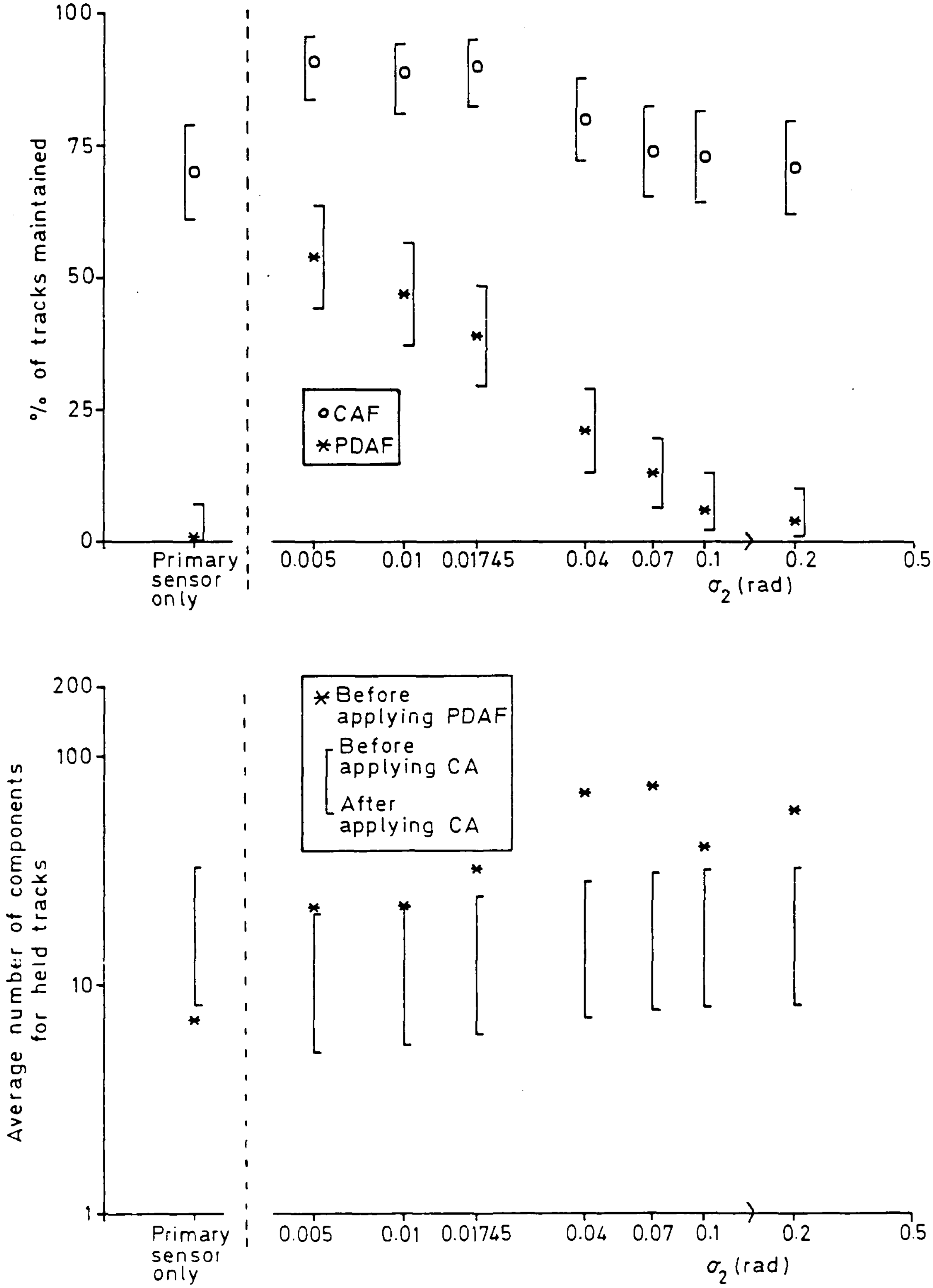


Fig 7.11 Auxiliary Sensor filter: ρ_2 = 45 rad^{-1}, $P_T = P_F = 0.9$ and σ_2 varies

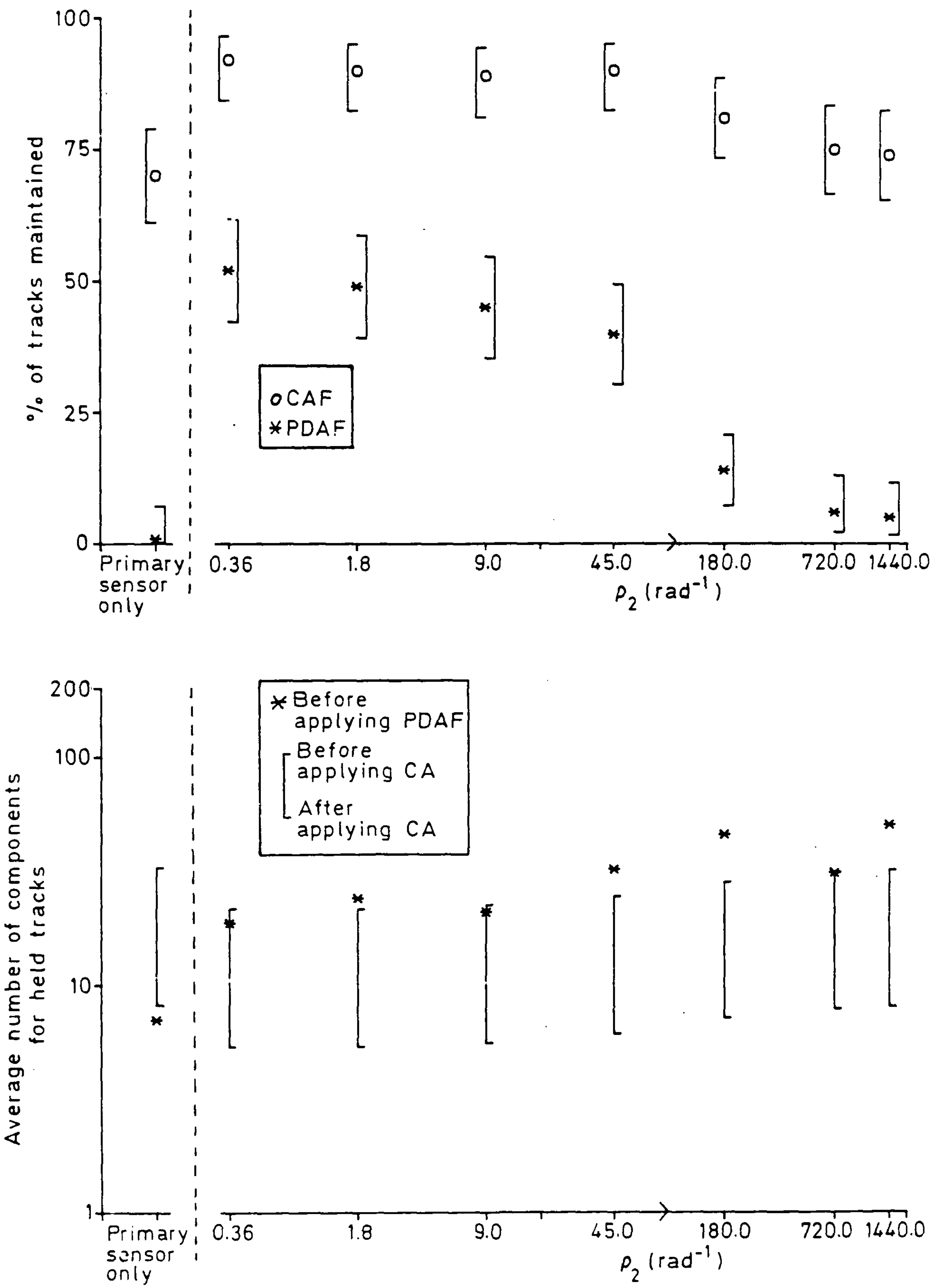


Fig 7.12 Auxiliary Sensor filter: σ_2 = 0.01745 radians, $P_T = P_F = 0.9$ and ρ_2 varies

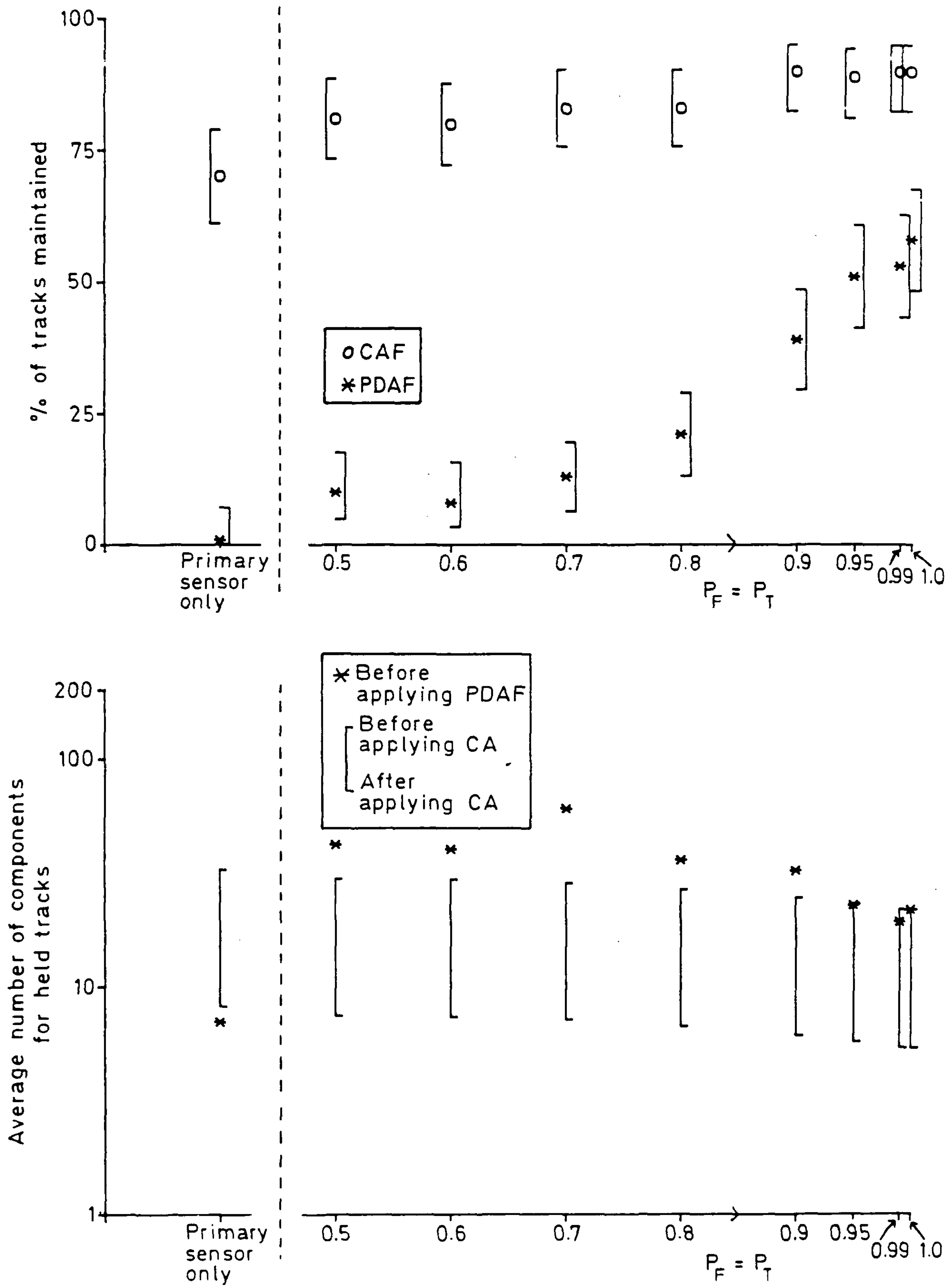


Fig 7.13 Auxiliary Sensor filter: σ_2 = 0.01745 rad, ρ_2 = 45 rad^{-1} and P_T varies

8 MULTIPLE MEASUREMENT CLASSES: THE PROBLEM OF INTERFERING MEASUREMENTS

8.1 Introduction

In the preceding chapters it has been assumed that measurements are either true or false, and that at most one of the measurements from a single sensor may be true at any time step. The problem is now extended to allow further classes of measurement which may or may not be associated with the target. The formal Bayesian solution to this new problem, which is given in the following section, is a straightforward extension of the baseline filter. However, except for simple cases, it is not easy to apply this general solution to derive practical filters for specific tracking examples. Thus to arrive at useful recursive filters it may be necessary to impose rather crude approximations.

In section 8.3 a tracking problem with three measurement classes is described. Two of these classes are the usual true and false measurements, while the third class consists of interfering measurements associated with the target position. As an extra complication this interference is intermittent and its switching on and off may be modelled by a Markov process. A practical sub-optimal tracking filter has been derived from the general solution of section 8.2 by making several approximations.

8.2 Problem formulation and general solution

At each time step a set of measurements Z is received:

$$Z = \left\{ \underline{z}_i : i = 1, m \right\} .$$

Each measurement $\underline{z}$ of Z may belong to any one of N_c classes, and the class membership of $\underline{z}$ may be unknown. However if $\underline{z}$ does belong to class C_j, then $\underline{z}$ is an independent sample from the pdf:

$$p\left(\underline{z}|\underline{x}\ ,\ C_j\right)\ , \tag{8.1}$$

which is assumed to be available. It is also assumed that the probability distribution of the number of received measurements from each class is given. Thus the probability of receiving m_j' measurements belonging to class j is known and is denoted $g_j(m_j')$. Note however that m_j' is in general not known and that the membership of each class may only be hypothesized. Clearly:

$$\sum_{j=1}^{N_c} m_j' = m\ .$$

As usual the state propagation equation is given by equation (2.1) and the problem is to obtain the posterior pdf of $\underline{x}$ at each time step.

To solve this problem, following section 2.3.2, it is necessary to construct all feasible measurement association hypotheses $\mathscr{H}'$ and so to evaluate the posterior pdf of $\underline{x}$:

$$p(\underline{x}|Z) = \sum_{\text{All}\,\mathscr{H}'} p(\underline{x}|\mathscr{H}',Z)\ \Pr\left\{\mathscr{H}'|Z\right\}\ . \tag{8.2}$$

(The time step subscript k and explicit dependency on $\mathscr{P}_k$ are omitted in the chapter, although the conditioning should be understood throughout.)

This equation is similar to equation (2.19) and as usual:

$$\mathscr{H}' = (\mathscr{H}, \Psi) \quad ,$$

is a joint hypothesis, where $\mathscr{H}$ is a hypothesis on the class membership of data received up to and including the previous time step, and Ψ is an association hypothesis on the current measurement set Z. Also we assume that $\Pr\left\{\mathscr{H}\right\}$ and $p(\underline{x}|\mathscr{H})$ are available from the previous recursion.

First consider $p(\underline{x}|\mathscr{H}',Z)$. From Bayes theorem:

$$p(\underline{x}|\mathscr{H}',Z) = \frac{p(Z|\underline{x},\mathscr{H}')\ p(\underline{x}|\mathscr{H}')}{p(Z|\mathscr{H}')} \quad . \tag{8.3}$$

Suppose that Ψ assigns the ith member of Z to class $C_{f(i)}$, then since the members of Z are independent:

$$p(Z|\underline{x},\mathscr{H}') = \prod_{i=1}^{m} p\left(\underline{z}_i|\underline{x},C_{f(i)}\right) \quad . \tag{8.4}$$

Also

$$p(\underline{x}|\mathscr{H}') = p(\underline{x}|\mathscr{H})$$

which is available from the previous recursion. The denominator of the RHS of equation (8.3) is given by:

$$\int p(Z|\underline{x},\mathscr{H}')\ p(\underline{x}|\mathscr{H}')\ \underline{dx} \quad . \tag{8.5}$$

Thus in principle $p(\underline{x}|\mathscr{H}',Z)$ can be found. In practice it is likely to be difficult to find a simple analytical expression unless the

underlying distributions are Gaussian or the measurements are independent of $\underline{x}$ for many classes (as in the baseline problem).

Now consider the posterior probability:

$$\Pr\left\{\mathscr{H}' \mid Z\right\} = \Pr\left\{\mathscr{H}, \Psi \mid Z\right\} .$$

This is given by equation (2.10):

$$\Pr\left\{\mathscr{H}' \mid Z\right\} = \frac{p(Z \mid \mathscr{H}') \ \Pr\{\Psi \mid \mathscr{H}\} \Pr\{\mathscr{H}\}}{p(Z)} . \tag{8.6}$$

Since the members of Z are independent, following equation (2.11) we have:

$$p(Z \mid \mathscr{H}') = \int p(Z \mid \underline{x}, \Psi) \ p(\underline{x} \mid \mathscr{H}) \ \underline{\underline{dx}}$$

$$= \int \prod_{i=1}^{m} p\left(\underline{z}_i \mid \underline{x}, C_{f(i)}\right) p(\underline{x} \mid \mathscr{H}) \ \underline{\underline{dx}} . \tag{8.7}$$

The factor $\Pr\left\{\Psi \mid \mathscr{H}\right\}$ is the prior probability of Ψ, and since this is independent of hypotheses on data from previous time steps:

$$\Pr\left\{\Psi \mid \mathscr{H}\right\} = \Pr\{\Psi\} .$$

The evaluation of this probability depends on what prior information is available on the class membership of the measurements Z. However it is known that the probability of receiving m_j' measurements belonging to class C_j is $g_j(m_j')$. Thus the joint prior probability that m_j' measurements belong to class C_j for $j = 1, \ldots, N_c$ is:

$$\prod_{j=1}^{N_c} g_j\left(m_j'\right) \quad ,$$

where $\sum_{j=1}^{N_c} m_j' = m$.

If there are no prior restrictions on the class membership of measurements, then the number of hypotheses Ψ that could have caused this distribution of measurements is:

$$\frac{m!}{m_1'!\ m_2'!\ \ldots\ m_{N_c}'!} \quad .$$

So, since *a priori* each of these hypotheses is equally probable:

$$\Pr\{\Psi\} = \frac{m_1'!\ m_2'!\ \ldots\ m_{N_c}'!}{m!} \prod_{j=1}^{N_c} g_j\left(m_j'\right) \quad , \tag{8.8}$$

where m_j' is the number of measurements assigned to class C_j under hypothesis Ψ . If the class membership of the measurements is restricted, there may be fewer hypotheses corresponding to this distribution of measurements, in which case equation (8.8) must be amended. The final factor $\Pr\left\{\mathscr{H}\right\}$ is available from the previous recursion, and the denominator of equation (8.6) is given by:

$$p(Z) = \sum_{\substack{\text{All}\\ \Psi}} \sum_{\substack{\text{All}\\ \mathscr{H}}} p(Z|\mathscr{H},\Psi)\ \Pr\{\Psi\}\ \Pr\left\{\mathscr{H}\right\} \quad . \tag{8.9}$$

Thus again all the required functions are available and in principle the required solution may be obtained by substituting into equation (8.6). The prediction forwards to obtain the prior pdf at the following time step follows from the propagation equation of the state vector, as indicated in section 2.3.3.

To show that this general solution may be reduced to the baseline problem, suppose that there are only two classes of measurements, true and false. At most one of the measurements may be true (class C_1) and the probability distribution of the number of true measurements is given by:

$$g_1(m_1') = \begin{cases} P_D^{m_1'} \left(1 - P_D\right)^{1 - m_1'} & \text{for} \quad m_1' = 0 \quad \text{or} \quad 1 \\ 0 & \text{otherwise} \quad . \end{cases}$$

...... (8.10)

Also from equation (2.2), a true measurement is a Gaussian distribution about $H\underline{x}$:

$$p\left(\underline{z}|\underline{x},C_1\right) = \mathcal{N}(\underline{z};H\underline{x},R) \quad . \tag{8.11}$$

False measurements (class C_2) are uniformly distributed and are independent of $\underline{x}$:

$$p\left(\underline{z}\,|\,\underline{x},C_2\right) = V^{-1} \quad , \tag{8.12}$$

where V is the volume of the sensor surveillance region. The probability of receiving m_2' false measurements is given by a Poisson distribution:

$$g_2\left(m_2'\right) = e^{-\rho V}\,(\rho V)^{m_2'}/m_2'! \qquad \text{for} \qquad m_2' \geq 0 \ . \tag{8.13}$$

As described in section 2.3.2, the hypothesis Ψ_j , for $j \neq 0$, indicates that $m_1' = 1$ and $m_2' = m-1$. So from equation (8.8), for $j \neq 0$:

$$\Pr\left\{\Psi_j\right\} = \frac{1!(m-1)!}{m!}\,g_1(1)g_2(m-1)$$

$$= \frac{1}{m}\,g_1(1)g_2(m-1) \quad . \tag{8.14}$$

If $j = 0$, then $m_1' = 0$ and $m_2' = m$, so from equation (8.8):

$$\Pr\left\{\Psi_0\right\} = \frac{0!m!}{m!}\,g_1(0)g_2(m)$$

$$= g_1(0)g_2(m) \quad . \tag{8.15}$$

By substituting equations (8.10) to (8.15) into the general solution given above, the solution of the baseline problem given in Chapter 2 may be obtained.

8.3 The sector scan problem with intermittent interference

8.3.1 Problem statement

In this extension of the sector scan problem (see section 6.2) interfering measurements may occur behind the target, when viewed from the sensor position at the origin. If the target position is (r, θ), then interfering measurements may occur in the region (see Fig 8.1):

$$\left.\begin{array}{c} r < \text{range} < r + r_I \\ \\ \theta - \theta_I < \text{bearing} < \theta + \theta_I \end{array}\right\} . \tag{8.16}$$

These measurements are uniformly distributed in polar co-ordinates at a density of ρ_I km^{-1} radians^{-1}, however they only occur within the surveillance sector. The switching on and off of the interference is a Markov process. Thus if the interference were present at time step k, the probability that it would be present at time step $k + 1$ is p_{11}, and the probability that it would not be present is $p_{10} = 1 - p_{11}$. Likewise the probability that there is no interference at time step $k + 1$ given there is none at time step k is p_{00}, while the probability of a transition from off to on is p_{01}.

In this example it is assumed that the parameters r_I, θ_I, ρ_I, p_{11} and p_{00} are all known and that interference is not present as the target enters the surveillance sector. Only one sensor is present at the origin and no classification information is available to distinguish between the true measurement, the interfering measurements and the usual false measurements.

The problem of estimating the state of abruptly changing systems has received considerable attention (see, for example, Tugnait[42], Tugnait and Haddad[43], Weiss *et al*[45] and Bolm[52]). As noted in section 6.4.3, abrupt target manoeuvres have been represented by allowing the equations of target motion to switch between different models. In this interference switching problem the target model is fixed, but the measurement environment may change suddenly according to the switching probabilities p_{10} and p_{01}. It is quite straightforward to incorporate this possible switching within the usual Bayesian framework, and this part of the solution (section 8.3.2.1) is similar to the development in the above references. However the updating of probabilities and pdfs on the assumption that interference is present is a new problem. We shall introduce approximations which allow a practical sub-optimal filter to be derived from the optimal solution.

8.3.2 Problem solution

8.3.2.1 Representation of intermittency

For this problem we have the measurement - class association hypothesis Ψ to consider as described in section 8.2, but in addition there is the uncertainty of whether or not interference is present. We introduce a variable γ which takes the value 1 if interference is present and 0 if it is not. This indicator only applies to the current set of measurements, previous hypotheses on the presence of interference being included in $\mathscr{H}$. Equation (8.2) which gives the posterior pdf of $\underline{x}$ should be extended to:

$$p(\underline{x} \mid Z) = \sum_{\gamma=0}^{1} \sum_{\substack{\text{All} \\ \mathscr{H}'}} p(\underline{x} \mid \gamma, \mathscr{H}', Z) \Pr\left\{\gamma, \mathscr{H}' \mid Z\right\} , \qquad (8.17)$$

where $\mathscr{H}' = (\Psi,\mathscr{H})$. From Bayes rule, the corresponding version of equation (8.6) is:

$$\Pr\left\{\gamma,\mathscr{H}'\,|\,Z\right\} = \frac{p(Z\,|\,\gamma,\mathscr{H},\Psi)\ \Pr\{\Psi\,|\,\gamma,\mathscr{H}\}\ \Pr\{\gamma\,|\,\mathscr{H}\}\ \Pr\{\mathscr{H}\}}{p(Z)} \quad . \quad (8.18)$$

In this expression the factor $\Pr\left\{\gamma|\mathscr{H}\right\}$ is given by the switching probabilities p_{ij} . So for instance, if under $\mathscr{H}$ interference were present at the previous time step and if $\gamma = 0$, then:

$$\Pr\left\{\gamma|\mathscr{H}\right\} = p_{10} \quad .$$

The other factors in expressions (8.17) and (8.18) are given by other equations in section 8.2.

8.3.2.2 <u>The likelihood of a set of measurements: $p(Z|x,\gamma,\Psi)$</u>

A key step in the solution of this problem is the evaluation of the likelihood $p(Z|\underline{x},\gamma,\Psi)$. The hypothesis Ψ assigns each member of Z to one of three classes. We define class 1 to be true measurements, class 2 to be the interfering measurements and class 3 to be the usual false measurements. If $\gamma = 0$, none of the measurements belong to class 2. For classes 1 and 3, we have as usual:

$$p\left(\underline{z}\,|\,\underline{x},C_1\right) = \mathcal{N}(\underline{z};\underline{h}(\underline{x}),R)$$

and

$$p\left(\underline{z}\,|\,\underline{x},C_3\right) = V^{-1} \quad ,$$

where V is the volume of the surveillance region. For class 2:

$$p\left(\underline{z}\,|\,\underline{x},C_2\right) = \left[H\left(r_m - r\right) - H\left(r_m - \left(r + r_I\right)\right)\right]$$

$$\times\left[H\left(\theta_m - \left(\theta - \theta_I\right)\right) - H\left(\theta_m - \left(\theta + \theta_I\right)\right)\right]/2\theta_I r_I \quad ,$$

where $H(.)$ is the Heavyside function,

(r,θ) is the target position in polar co-ordinates, and:

$$\underline{z} = \begin{pmatrix} r_m \\ \\ \theta_m \end{pmatrix}.$$

The pdf takes this form because class 2 measurements cannot lie outside the interference region (which is assumed to be within the surveillance region for this expression).

Suppose that under hypothesis Ψ for the measurements received at a particular time step, measurement t belongs to class 1, m_2' measurements with subscripts from the set $\mathbb{C}_2$ belong to class 2 and the m_3' remaining measurement belong to class 3. In this case the likelihood of the recieved measurements Z is given by (from equation (8.4)):

$$p(Z\,|\,\underline{x},\gamma,\Psi) = V^{-m_3'}\,\mathcal{N}\left(\underline{z}_t;\underline{h}(\underline{x}),R\right)\prod_{i\in\mathbb{C}_2} p\left(\underline{z}_i\,|\,\underline{x},C_2\right) \quad .$$

If none of the measurements is true, the factor $\mathcal{N}(\underline{z}_t;\underline{h}(\underline{x}),R)$ is omitted. By considering the factors in the product of the class 2 terms it it can be seen that:

$$\prod_{i\in\mathbb{C}_2} p\left(\underline{z}_i|\underline{x},C_2\right) = \begin{cases} 0, \text{ if } r_{MX} - r_{MN} > r_I \text{ or } \theta_{MX} - \theta_{MN} > 2\theta_I \\ \left[H\left(r - r_{MX} + r_I\right) - H\left(r - r_{MN}\right)\right] \\ \times\left[H\left(\theta - \theta_{MX} + \theta_I\right) - H\left(\theta - \theta_{MN} - \theta_I\right)\right]\left(\frac{1}{2\theta_I r_I}\right)^{m_2'} , \\ \qquad \text{otherwise} \end{cases}$$

...... (8.19)

where r_{MX} and r_{MN} are the maximum and minimum range measurements in class 2, and θ_{MX} and θ_{MN} are the maximum and minimum bearing measurements in class 2. Note that expression (8.19) is sensible because if two measurements allocated to class 2 by Ψ are separated in range by more than r_I or in bearing by more than $2\theta_I$, the hypothesis must be false. If this is not so, the extreme class 2 measurements restrict the possible target position under Ψ to the rectangle in r,θ space shown in Fig 8.2. This is equivalent to a region A in x,y space, and for convenience we shall define the function:

$$U_A(\underline{x}) = \begin{cases} 0, \text{ if } m_2' > 0 \text{ and } \left\{x^2 + y^2 > r_{MN}^2 \right. \\ \quad \text{or } x^2 + y^2 < \left(r_{MX} - r_I\right)^2 \text{ or } \tan^{-1}(y/x) > \theta_{MN} + \theta_I \\ \quad \left. \text{or } \tan^{-1}(y/x) < \theta_{MX} - \theta_I \right\} \\ 0, \text{ if } A \text{ does not exist but } m_2' > 0 \\ 1, \text{ otherwise (including the case } m_2' = 0) . \end{cases}$$

Thus the required likelihood of the measurements Z is given by:

$$p(Z|\underline{x},\gamma,\Psi) = V^{-m_3'}\left(\frac{1}{2\theta_I r_I}\right)^{m_2'} U_A(\underline{x}) \begin{cases} \mathcal{N}\left(\underline{z}_t;\underline{h}(\underline{x}),R\right) & \text{if} \quad m_1' = 1 \\ \\ 1 \quad \text{if} \quad m_1' = 0 \end{cases} \qquad \ldots\ldots (8.20)$$

8.3.2.3 <u>First approximation: the prior pdf $p(\underline{x}|\mathscr{H}')$ is Gaussian</u>

Having found this likelihood we may proceed with the solution via equations (8.3) and (8.18). However to arrive at a practical filter it will be necessary to make a number of simplifying assumptions so that the resulting filter is sub-optimal. The first approximation is that the prior pdf $p(\underline{x}|\mathscr{H}')$ in equation (8.3) is Gaussian:

$$p(\underline{x}|\mathscr{H}') = \mathcal{N}\left(\underline{x};\bar{\underline{x}}_i,M_i\right) \quad ,$$

where i refers to the hypothesis $\mathscr{H}$. As will be seen from equation (8.22) below, this is incorrect, but it allows us to write:

$$\begin{aligned} p\left(\underline{x}|\underline{z}_t,\mathscr{H}'\right) &\propto p\left(\underline{z}_t|\underline{x},\mathscr{H}'\right) p(\underline{x}|\mathscr{H}') \\ &= \mathcal{N}\left(\underline{z}_t;\underline{h}(\underline{x}),R\right)\mathcal{N}\left(\underline{x};\bar{\underline{x}}_i,M_i\right) \\ &\simeq \mathcal{N}\left(\underline{x};\hat{\underline{x}}_{it},P_{it}\right)\mathcal{N}\left(\underline{z}_t;\underline{h}\left(\bar{\underline{x}}_i\right),S_i\right) \quad , \qquad (8.21) \end{aligned}$$

where we have made use of the extended Kalman filter approximation (see section 6.2). Thus, using the above result with (8.20) in equation (8.3), we obtain:

$$p(\underline{x}|\gamma,\mathscr{H}',Z) = \begin{cases} U_A(\underline{x})\mathcal{N}\left(\underline{x};\underline{\hat{x}}_{it},P_{it}\right)/F_1 & \text{if} \quad m_1' = 1 \\ U_A(\underline{x})\mathcal{N}\left(\underline{x};\underline{\bar{x}}_i,M_i\right)/F_2 & \text{if} \quad m_1' = 0 \ , \end{cases} \qquad \text{...... (8.22)}$$

where $F_1 = \int U_A(\underline{x})\mathcal{N}\left(\underline{x};\underline{\hat{x}}_{it},P_{it}\right)\underline{dx}$ and F_2 is similar. The function $U_A(\underline{x})$ effectively truncates the Gaussian in (8.22), so that the uncertainty in the value of $\underline{x}$ is reduced by the information from the class 2 measurements. Unfortunately the integrals F_1 and F_2 of the Gaussian over the region A cannot be evaluated analytically. However if $(\hat{x}_{it}, \hat{y}_{it})$ were well inside A and the corresponding standard deviations from P_{it} were small compared with the dimensions of A, the effect of $U_A(\underline{x})$ in equation (8.22) could be ignored.

Now consider the posterior probability of hypothesis $(\gamma,\mathscr{H}')$, given by equation (8.18). The probability of receiving m_2' interfering class 2 measurements is given by a Poisson distribution with mean $\rho_I V_I$, where $V_I = 2\theta_T r_I$ is the volume of the interference region. Similarly the probability of receiving m_3' false measurements is given by a Poisson distribution with mean ρV, where V is the volume of the surveillance region. Thus from equation (8.8):

$$\Pr\left\{\Psi|\gamma,\mathscr{H}\right\} = \frac{P_D^{m_1'}\left(1 - P_D\right)^{m_1'-1}}{m!} e^{-\rho V}(\rho V)^{m_3'} \begin{cases} e^{-\rho_I V_I}\left(\rho_I V_I\right)^{m_2'}, & \text{if } \gamma = 1 \\ 1 & , \text{if } \gamma = 0 \end{cases}$$

...... (8.23)

Also using (8.20) and (8.21) with equation (8.7), we have:

$$p(Z|\gamma,\mathscr{H}') = \begin{cases} V^{-m_3'} V_I^{-m_2'} \mathscr{N}\left(\underline{z}_t;\underline{h}\left(\bar{\underline{x}}_i\right),S_i\right)\int U_A(\underline{x})\mathscr{N}\left(\underline{x};\hat{\underline{x}}_{it},P_{it}\right)\underline{dx} & \text{if} \quad m_1' = 1 \\ V^{-m_3'} V_I^{-m'_2} \int U_A(\underline{x})\mathscr{N}\left(\underline{x};\bar{\underline{x}}_i,M_i\right)\underline{dx} & \text{if} \quad m_1' = 0 \end{cases}$$

...... (8.24)

Inserting (8.23) and (8.24) into (8.18) we obtain:

$$\Pr\left\{\gamma,\mathscr{H}'|Z\right\} = \begin{cases} P_D\, e^{-\rho V} \rho^{m_3'} \mathscr{N}\left(\underline{z}_t;\underline{h}\left(\bar{\underline{x}}_i\right),S_i\right) \Pr\left\{\gamma|\mathscr{H}\right\} \Pr\left\{\mathscr{H}\right\} \\ \times \left\{ \begin{array}{ll} e^{-\rho_I V_I} \rho_I^{m_2'} \int U_A(\underline{x})\mathscr{N}\left(\underline{x};\hat{\underline{x}}_{it},P_{it}\right)\underline{dx}, & \text{if } \gamma = 1 \\ 1 & , \text{if } \gamma = 0 \end{array} \right\} / E \\ \text{if} \quad m_1' = 1 \end{cases}$$

$$\Pr\left\{\gamma,\mathscr{H}'\middle|Z\right\} = \begin{cases} \left(1-P_D\right) e^{-\rho V} \rho^{m_3'} \Pr\left\{\gamma\middle|\mathscr{H}\right\} \Pr\left\{\mathscr{H}\right\} \\ \times \left\{\begin{array}{ll} e^{-\rho_I V_I} \rho_I^{m_2'} \int U_A(\underline{x}) \mathscr{N}\left(\underline{x};\bar{\underline{x}}_i,M_i\right)\underline{dx}, & \text{if } \gamma = 1 \\ 1 & , \text{if } \gamma = 0 \end{array}\right\} / E \\ \text{if} \quad m_1' = 0 \end{cases} \qquad \ldots\ldots (8.25)$$

where E is the normalizing denominator which is chosen so that:

$$\sum_{\gamma=0}^{1} \sum_{\substack{\text{All} \\ \mathscr{H}'}} \Pr\left\{\gamma,\mathscr{H}'\middle|Z\right\} = 1 \quad .$$

Thus in principle, the posterior pdf of $\underline{x}$ may be obtained by substituting (8.22) and (8.25) into (8.17) and summing over all feasible hypotheses. The main difficulties here are that an integral of the form:

$$\int U_A(\underline{x}) \mathscr{N}\left(\underline{x};\hat{\underline{x}}_{it},P_{it}\right)\underline{dx}$$

must be evaluated for every hypothesis with $\gamma = 1$ and that there are a very large number of feasible hypotheses. In fact, if m measurements are received and if $\gamma = 1$, the number of feasible hypotheses Ψ concerning the class membership of measurements is $(2 + m)2^{m-1}$. This figure is very large even for modest values of m : for $m = 20$ there are over 10^7 measurement association hypotheses. So, to derive a practical filter further simplifying approximations must be introduced.

8.3.2.4 Further approximations to derive a practical filter

Firstly we shall ignore the contribution of the class 2 measurements in the expression for $p(\underline{x}|\gamma,\mathscr{H}',Z)$. Thus equation (8.22) becomes:

$$p(\underline{x}|\gamma,\mathscr{H}',Z) = \begin{cases} \mathscr{N}\left(\underline{x};\hat{\underline{x}}_{it},P_{it}\right) & \text{if} \quad m_1' = 1 \\ \mathscr{N}\left(\underline{x};\bar{\underline{x}}_i,M_i\right) & \text{if} \quad m_1' = 0 \ . \end{cases}$$

...... (8.26)

(This is the same sort of approximation as made in the deviation of the Auxiliary Sensor filter (see section 7.4.2).) Thus information from the class 2 measurements is only taken into account via the probabilities $\Pr\{\gamma,\mathscr{H}'|Z\}$. Clearly some potentially valuable information is being discarded here, however it does allow a useful simplification of equation (8.17) (and it ensures that the prior pdf $p(\underline{x}|\mathscr{H}')$ is Gaussian). Let us write the measurement association hypothesis Ψ as the pair:

$$\Psi = (\Omega,\Lambda) \quad ,$$

where Ω indicates the choice of true measurement (class 1) and Λ specifies the partition of the remaining measurements between classes 2 and 3. From (8.26) it can be seen that $p(\underline{x}|\gamma,\mathscr{H}',Z)$ is independent of Λ , so that equation (8.17) may be written:

$$p(x|Z) = \sum_{\gamma=0}^{1} \sum_{\substack{\text{All}\\ \mathscr{H}}} \sum_{\substack{\text{All}\\ \Omega}} p(\underline{x}|\gamma,\Omega,\mathscr{H},Z)\ \Pr\left\{\gamma,\Omega,\mathscr{H}|Z\right\} \tag{8.27}$$

where $p(\underline{x}|\gamma,\Omega,\mathscr{H},Z)$ is given by equation (8.26)

and

$$\Pr\left\{\gamma,\Omega,\mathscr{H}|Z\right\} = \sum_{\substack{\text{All}\\ \Lambda}} \Pr\left\{\gamma,\Omega,\Lambda,\mathscr{H}|Z\right\} \quad . \tag{8.28}$$

The usual acceptance test may be employed to make a short list of Ω hypotheses for each hypothesis $\mathscr{H}$.

We now introduce the last approximation which allows us to perform the summation over all Λ in equation (8.28). It is assumed, *only* for the integrals in equation (8.25), that:

$$\left.\begin{aligned} \mathcal{N}\left(\underline{x};\underline{\hat{x}}_{it},P_{it}\right) &= \delta\left(\underline{x} - \underline{\hat{x}}_{it}\right) \\ \text{and} \qquad \mathcal{N}\left(\underline{x};\underline{\bar{x}}_{i},M_{i}\right) &= \delta\left(\underline{x} - \underline{\bar{x}}_{i}\right) \end{aligned}\right\} \tag{8.29}$$

Thus the uncertainty in the value of $\underline{x}$ represented by P_{it} or M_i under the hypothesis $(\Omega,\mathscr{H})$ is ignored. It is recognized that this contradicts the first assumption given by equation (8.26), and although this is unsatisfactory, it does enable a practical filter to be derived.

Consider the probability $\Pr\{\gamma,\Omega,\Lambda,\mathscr{H}|Z\}$ given by equation (8.25) in the light of assumption (8.29). With this assumption, for $\gamma = 1$, the interference region is known precisely under hypothesis $(\Omega,\mathscr{H})$.

Thus if all of the measurements associated with class 2 by hypothesis Λ are in this region, then the integral in (8.25) is unity. Otherwise the integral is zero, so that the probability of this hypothesis is zero. So, from (8.25):

$$
\Pr\left\{\gamma,\Omega,\Lambda,\mathscr{H}\,|\,Z\right\} = \begin{cases}
e^{-\rho V}\,\rho^{m_3'}\,e^{-\rho_I V_I}\,\rho_I^{m_2'}\,\Pr\left\{\gamma\,|\,\mathscr{H}\right\}\Pr\left\{\mathscr{H}\right\} \\
\times \begin{Bmatrix} P_D\,\mathcal{N}\left(\underline{z}_t;\underline{h}\left(\bar{\underline{x}}_i\right),S_i\right), & \text{if} & m_1' = 1 \\ 1 - P_D, & \text{if} & m_1' = 0 \end{Bmatrix} \Big/ E \\
\text{if } \gamma = 1 \text{ and if under } \Lambda\text{, all of the } m_2' \\
\text{class 2 measurements are within the interference} \\
\text{region defined by } \Omega\,. \\
\\
e^{-\rho V}\,\rho^{m_3'}\,\Pr\left\{\gamma\,|\,\mathscr{H}\right\}\Pr\left\{\mathscr{H}\right\} \\
\times \begin{Bmatrix} P_D\,\mathcal{N}\left(\underline{z}_t;\underline{h}\left(\bar{\underline{x}}_i\right),S_i\right), & \text{if} & m_1' = 1 \\ 1 - P_D, & \text{if} & m_1' = 0 \end{Bmatrix} \Big/ E \\
\text{if } \gamma = 0\,. \\
0, \text{ otherwise.}
\end{cases}
\qquad \text{...... (8.30)}
$$

Now if $\gamma = 0$, then all measurements apart from the true measurement belong to class 3, *ie* if $\gamma = 0$, for each $(\Omega,\mathscr{H})$ there is only one hypothesis Λ . So if $\gamma = 0$, there is only one term in the summation of equation (8.28) and $\Pr\{\gamma,\Omega,\mathscr{H}|Z\}$ is given directly by equation (8.30). However if $\gamma = 1$, then for each $(\Omega,\mathscr{H})$, the number of feasible hypotheses Λ is equal to the number of ways of partitioning the measurements in the interference region between classes 2 and 3. Suppose that with $\gamma = 1$, under hypothesis $(\Omega,\mathscr{H})$, m_I measurements fall within the interference region. If m_2' of these measurements belong to class 2, then there are exactly $\binom{m_I}{m_2'}$ ways of partitioning the m_I measurements between classes 2 and 3. Since the measurements outside the interference region all belong to class 3 (excluding the true measurement), there are exactly $\binom{m_I}{m_2'}$ feasible hypotheses Λ for which m_2' measurements belong to class 2. The probability of each of these hypotheses is the same and it is given by (8.30). Also since m_2' may take any value between 0 and m_I , for $\gamma = 1$ the summation (8.28) may be evaluated using equation (8.30) and the identity:

$$\sum_{m_2'=0}^{m_I} \binom{m_I}{m_2'} \rho_I^{m_2'} \rho^{m_I - m_2'} = \left(\rho + \rho_I\right)^{m_I} .$$

Thus the probability $\Pr\{\gamma,\Omega,\mathscr{H}|Z\}$ may be written (absorbing some common factors into the normalizing denominator):

$$
\Pr\left\{\gamma,\Omega,\mathscr{H}\mid Z\right\} = \begin{cases}
\left(1+\frac{\rho_I}{\rho}\right)^{m_I} e^{-\rho_I V_I} \Pr\left\{\gamma\mid\mathscr{H}\right\} \Pr\left\{\mathscr{H}\right\} \\
\times \left\{\begin{bmatrix} P_D \mathcal{N}\left(\underline{z}_t;\underline{h}\left(\bar{\underline{x}}_i\right),S_i\right), & \text{if} \quad m_1' = 1 \\ \left(1-P_D\right)\rho \quad , & \text{if} \quad m_1' = 0 \end{bmatrix}\right\} / E \\
\text{if} \quad \gamma = 1 \\
\Pr\left\{\gamma\mid\mathscr{H}\right\} \Pr\left\{\mathscr{H}\right\} \\
\times \left\{\begin{bmatrix} P_D \mathcal{N}\left(\underline{z}_t;\underline{h}\left(\bar{\underline{x}}_i\right),S_i\right), & \text{if} \quad m_1' = 1 \\ \left(1-P_D\right)\rho \quad , & \text{if} \quad m_1' = 0 \end{bmatrix}\right\} / E \\
\text{if} \quad \gamma = 0
\end{cases}
$$

...... (8.31)

where the normalizing denominator E is chosen so that:

$$
\sum_{\gamma=0}^{1} \sum_{\substack{\text{All} \\ \Omega}} \sum_{\substack{\text{All} \\ \mathscr{H}}} \Pr\left\{\gamma,\Omega,\mathscr{H}\mid Z\right\} = 1 \quad .
$$

By applying these approximations, the number of hypotheses that must be explicitly considered has been reduced to a feasible number, provided that the usual acceptance test and mixture reduction algorithm

are employed. The posterior probabilities for the feasible hypotheses are simple modifications to those of the baseline problem given by equation (2.18). To evaluate the main modifying factor for $\gamma = 1$, it is only necessary to count the number m_I of measurements falling within the interference region defined by the hypothesis $(\Omega,\mathscr{H})$; the awkward integrals of expression (8.25) are avoided. Due to the incorrect assumption (8.29) that the state vector is perfectly known under $(\Omega,\mathscr{H})$, this modifying factor may be overselective, so occassionally an undue weighting is given to the wrong component. To compensate for this, in the evaluation of m_I a heuristic adjustment has been made to the boundaries of the interference region as defined by $(\Omega,\mathscr{H})$. Each azimuth boundary has been increased by one standard deviation of the true measurement bearing error σ_θ, and each range boundary has been increased by σ_r (see Fig 8.3). This has the effect of 'softening' the selectivity of the modifying factor $(1 + \rho_I/\rho)^{m_I}$ in (8.31). Further details of the filter implementat'`n are described in the following section.

8.3.3 Implementation of the filter

The implementation of the tracking filter derived in the previous sections is based on equations (8.26), (8.27) and (8.31). The formation and control of hypotheses is shown schematically in Fig 8.4. Each hypothesis $\mathscr{H}$ from the previous time step is predicted forwards and the usual acceptance test is applied to identify a set of probable true measurements for each $\mathscr{H}$. Together with the possibility that the true measurement has been missed, these sets make up the Ω hypotheses. For each $(\Omega,\mathscr{H})$ hypothesis, the posterior pdf of $\underline{x}$ is evaluated from equation (8.26) (from our approximation this is independent of γ). Each $(\Omega,\mathscr{H})$ hypothesis is then split to allow

for the possibilities of interference absent or present ($\gamma = 0$ or 1), and the posterior probability of each $(\gamma,\Omega,\mathscr{H})$ hypothesis is calculated from equation (8.31).

The mixture components and probability weights of the posterior pdf of $\underline{x}$ for the current time step are now available (see equation (8.27)). The required estimate may now be extracted, the usual minimum mean square estimate being given by:

$$\hat{\underline{x}} = \sum_{\substack{\text{All}\\ \mathscr{H}}} \sum_{\substack{\text{All}\\ \Omega}} \left(\sum_{\gamma=0}^{1} \Pr\left\{\gamma,\Omega,\mathscr{H} \mid Z\right\} \right) \begin{Bmatrix} \hat{\underline{x}}_{it}, & \text{if} & m_1 = 1 \\ \bar{\underline{x}}_{i}, & \text{if} & m_1 = 0 \end{Bmatrix} \qquad \ldots\ldots (8.32)$$

where $\hat{\underline{x}}_{it}$ and $\bar{\underline{x}}_{i}$ are the means of the mixture components (see equation (8.26)), subscript i corresponds to $\mathscr{H}$, and t is the choice of true measurement defined by Ω . Also the probability that interference is present, based on the filter's processing of the received measurements, is given by:

$$P_I = \sum_{\substack{\text{All}\\ \mathscr{H}}} \sum_{\substack{\text{All}\\ \Omega}} \Pr\left\{\gamma = 1,\Omega,\mathscr{H} \mid Z\right\} . \qquad (8.33)$$

For implementation using the Clustering Algorithm, before mixture reduction the hypotheses are divided into two groups for $\gamma = 0$ and $\gamma = 1$. The Clustering Algorithm is then applied separately to the mixture distribution corresponding to each group. This ensures that even after reduction, each mixture component is associated with

$\gamma = 0$ or $\gamma = 1$. The reduced mixture can then be predicted forwards in the usual way, ready for the next set of measurements.

An implementation using a PDAF type of approximation is shown schematically in Fig 8.5. This is slightly different from the usual PDAF philosophy in that two mixture components are allowed to survive at each time step. These two components correspond to interference present or interference absent. When a set of measurements is received, each of these components is split according to $\gamma = 1,0$, and the PDAF is applied to each branch. Thus four branches are created with probability weights β_{00}, β_{01}, β_{10} and β_{11} (see Fig 8.5). The two $\gamma = 0$ branches and the two $\gamma = 1$ branches are then merged separately to form a two component mixture distribution with probability weights:

$$\beta_1 = \beta_{01} + \beta_{11}$$

and

$$\beta_0 = \beta_{00} + \beta_{10} \quad .$$

These components are predicted forwards to the next time step.

Note that the standard PDAF avoids the calculation of the mean of each mixture component before reduction. However for this problem, as for the Auxiliary Sensor filter (section 7.4.2), to evaluate the required probability weights, the means must be available. They are required to identify the interference region so that m_I can be found (see (8.31)). Thus much of the efficiency of the standard PDAF is lost in this implemenation.

8.3.4 Simulation example

Target trajectories and measurements have been simulated for the problem defined in section 8.3.1. The standard sector scan parameters shown in Table 6.1 have been used with the following interference parameters:

$$\text{Interference region} \begin{cases} r_I = 5 \text{ km} \\ \theta_I = 0.04 \text{ radians} \end{cases}$$

$$\text{Switching probabilities} \begin{cases} p_{11} = 0.9,\ p_{10} = 0.1 \\ p_{00} = 0.9,\ p_{01} = 0.1\ . \end{cases}$$

One hundred replications of trajectories and measurements have been generated for each of the following values of interfering measurement density ρ_I: 10, 20, 40, 100, 200 and 400 $km^{-1}rad^{-1}$. (Note that the density of the usual false measurements is $\rho = 10\ km^{-1}rad^{-1}$.) The standard sector scan filter, which assumes there is no interference, and the Interference filter described in the previous section have both been applied to the simulated data. In each case results have been obtained for both PDAF and Clustering Algorithm reduction techniques. The percentage of maintained tracks for each of these filters is shown in Fig 8.6 as a function of ρ_I .

The introduction of intermittent interference with $\rho_I = 10\ km^{-1}$ rad^{-1} has negligible effect on the performance of the standard CAF

and PDAF. Also the performance of the Interference CAF is very similar to that of the standard CAF, and likewise the performance of both PDAFs is similar. (For this low level of ρ_I, with interference switched on, the average number of interfering measurements generated per scan is only four.) With increasing ρ_I, the percentage of maintained tracks for the standard filters tends to decrease, as would be expected. However the performance of the Interference PDAF improves with ρ_I and tends towards the performance of the Interference CAF. This improvement is because the Interference PDAF is making use of information from the interfering measurements. As ρ_I increases, more measurements fill out the interference region, so that the boundaries of the region become more well defined (see Fig 8.2). Thus the probability weight for the correct Ω hypothesis is more strongly reinforced as ρ_I increases. The percentage of tracks held by the Interference CAF remains roughly constant at about 95% as ρ_I increases; by modelling the intermittent interference, the performance degradation of the standard CAF is avoided.

As expected, the average number of mixture components generated increases with ρ_I for both standard and Interference filters (see Fig 8.6). For $\rho_I \leqslant 40\ \mathrm{km}^{-1}\mathrm{rad}^{-1}$, the standard filters generate less components than the Interference filters, while for $\rho_I \geqslant 100\ \mathrm{km}^{-1}\mathrm{rad}^{-1}$, the standard filters generate more components. This may be explained as follows. When ρ_I is small, the standard filters are only likely to encounter a few interfering measurements, especially if track is maintained. If the Interference filters encounter a similar measurement density, they will generate more components since allowance is made for the possibilities $\gamma = 0$ and $\gamma = 1$. When ρ_I is large, even if the standard filters maintain track, they are likely to have been attracted

into an interference region (and so encountered a high density of measurements) during their traversal of the sector. However the Interference filters recognize that interference may be present, and make use of their knowledge of the distribution of these measurements relative to the target position, to lead the track beside the interference region. Thus these filters avoid regions of high measurement density and so for large ρ_I, on average they generate less components than the standard filters, even though the possibilities $\gamma = 0$ and $\gamma = 1$ are included.

The variation with ρ_I in the number of components generated by the CAF is reflected in the average cpu time to perform a single iteration (see Table 8.1). However the processing time for the standard PDAF is always less than the Interference PDAF, which requires explicit calculation of the mean of each mixture component (see section 8.3.3). The error statistic $\bar{\bar{E}}$ is also given in Table 8.1. This shows that even for held tracks, both the standard and the Interference filters tend to underestimate their tracking error, particularly for large values of ρ_I. This is probably due to the rather sweeping approximations made in the deviation of the Interference filters and the omission of any interference model for the standard filters. The error underestimate is worst for the standard PDAF. As for the standard sector scan problem, for lost tracks the filters often seriously underestimate the tracking error.

Fig 8.7 shows the filter's achieved mean square position error over the first forty time steps for $\rho_I = 40$ and $400\ \mathrm{km}^{-1}\mathrm{rad}^{-1}$. For maintained tracks, the accuracy of the Interference filters is superior to the standard filters. The improvement is most evident for $\rho_I = 400\ \mathrm{km}^{-1}\mathrm{rad}^{-1}$. For the standard filter, tracking error for the

held tracks increases with ρ_I, while for the Interference filters the CAF is little changed and the PDAF shows some improvement.

Finally, Figs 8.8 to 8.10 show three examples of target tracking for $\rho_I = 10$, 40 and 400 $km^{-1}rad^{-1}$. Each of these figures shows trajectory estimates produced by the CAF and the PDAF for both the standard filter and the Interference filter. The points at which interference switches on and off are indicated on the actual target paths. Also for each example a plot of interference switching against time is presented. This may be compared with the Interference filters' internal assessment of the probability P_I that interference is present (obtained by summing over the appropriate mixture weights - see equation (8.33)). For $\rho_I = 400\ km^{-1}rad^{-1}$, the plots of P_I are essentially identical to the actual switching waveform, showing that the filters are very certain as to the presence or absence of interference. With this high density it is easy for the filters to detect the large number of extra measurements behind the target when interference is present. (A sample plot of the measurements received on a single scan for $\rho_I = 400\ km^{-1}rad^{-1}$ is shown in Fig 8.11.) As ρ_I is reduced the presence or absence of interference becomes more difficult to detect, and for the sparse interference $\rho_I = 10\ km^{-1}rad^{-1}$, the traces of P_I are quite different from the actual switching signal (see Fig 8.8). At the higher densities of $\rho_I = 40$ and 400 $km^{-1}rad^{-1}$, the effect on the standard filter tracks of interference appearing behind the target is obvious, and the value of modelling the interference is clearly demonstrated.

8.4 Conclusions

It is fairly straightforward to derive the formal Bayesian solution to the extension of the baseline problem to multiple

measurement classes. However the interference example shows that very complex filters may result when this general solution is applied to specific problems. By making several approximations a practical filter has been derived for the interference problem. In spite of these approximations, simulations show the performance benefit of modelling the intermittent interference. Especially for high levels of interference, the performance of the multiple measurement class filter is clearly superior to the standard filter which takes no account of possible interference.

Table 8.1

PROCESSOR TIMINGS AND ERROR STATISTIC $\bar{E}$ FOR SECTOR SCAN PROBLEM WITH INTERMITTENT INTERFERENCE

Density of interfering measurements ρ_I (km^{-1} rad^{-1})	Tracks held or lost	Standard filters				Interference filters			
		Average cpu time for single step (ms)		Error statistic $\bar{E}$		Average cpu time for single step (ms)		Error statistic $\bar{E}$	
		CAF	PDAF	CAF	PDAF	CAF	PDAF	CAF	PDAF
No interference	H	1.68	0.477	3.912	3.767	3.15	1.09	3.908	3.767
	L	1.64*	0.669	1648.000*	637.600	2.96	8.37	1645.000	711.400
10.0	H	1.97	0.482	3.975	10.340	3.57	1.16	4.299	4.206
	L	1.47*	0.583	2952.0*	2714.000	3.51*	1.83	1239.0*	1693.000
20.0	H	2.17	0.495	4.182	7.132	3.66	1.18	4.050	4.809
	L	1.93	0.645	2961.000	625.400	3.64	1.56	4054.000	1975.000
40.0	H	3.06	0.506	4.079	6.158	4.16	1.24	4.136	9.071
	L	3.71	0.613	11790.000	461.500	4.06	1.23	6372.000	1789.000
100.0	H	13.80	0.555	6.231	32.230	5.89	1.38	6.510	13.310
	L	16.60	0.666	7528.000	199.700	4.76	1.15	1224.000	1742.000
200.0	H	38.00	0.644	4.906	47.450	9.08	1.64	4.665	5.580
	L	48.90	0.675	1821.000	3325.000	7.81*	1.09	1096.0*	2533.000
400.0	H	61.80	0.728	4.651	12.040	11.20	2.25	8.485	8.397
	L	74.40	0.791	2942.000	559.400	7.86	1.78	804.400	1864.000

* Indicates a small sample (less than five replications)

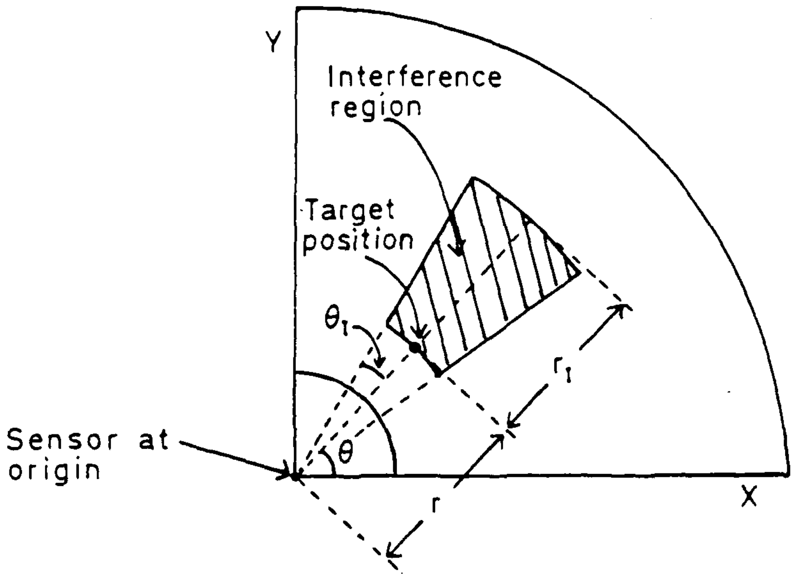


Fig 8.1 The interference region

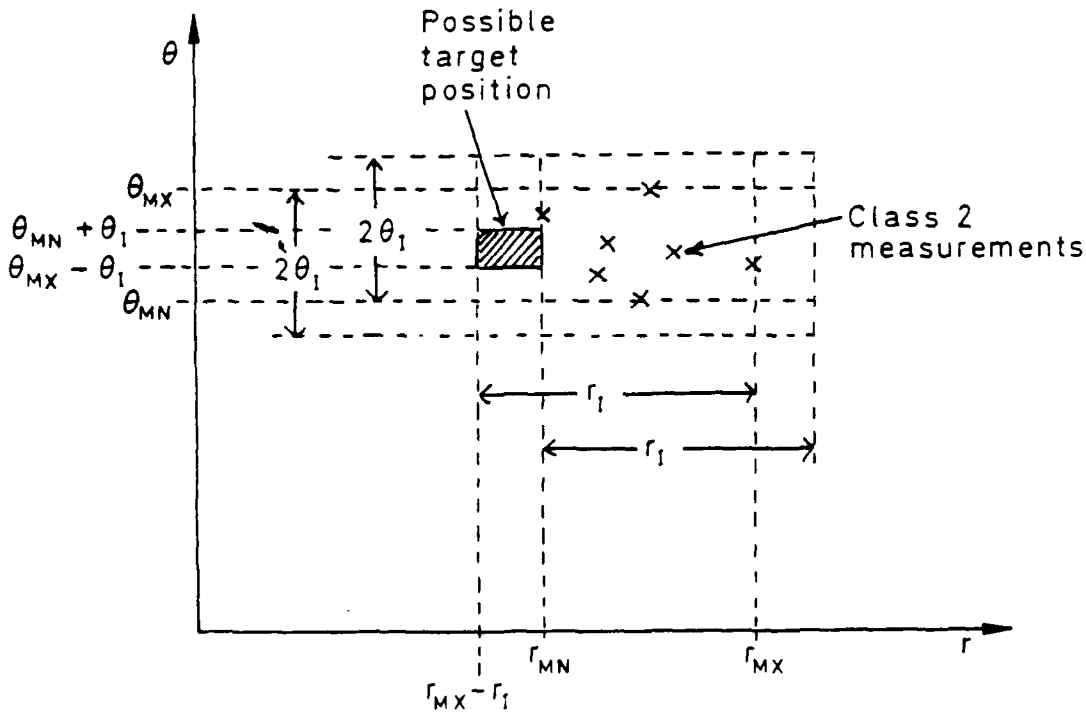


Fig 8.2 Given a set of class 2 measurements, the target must lie inside the shaded rectangle

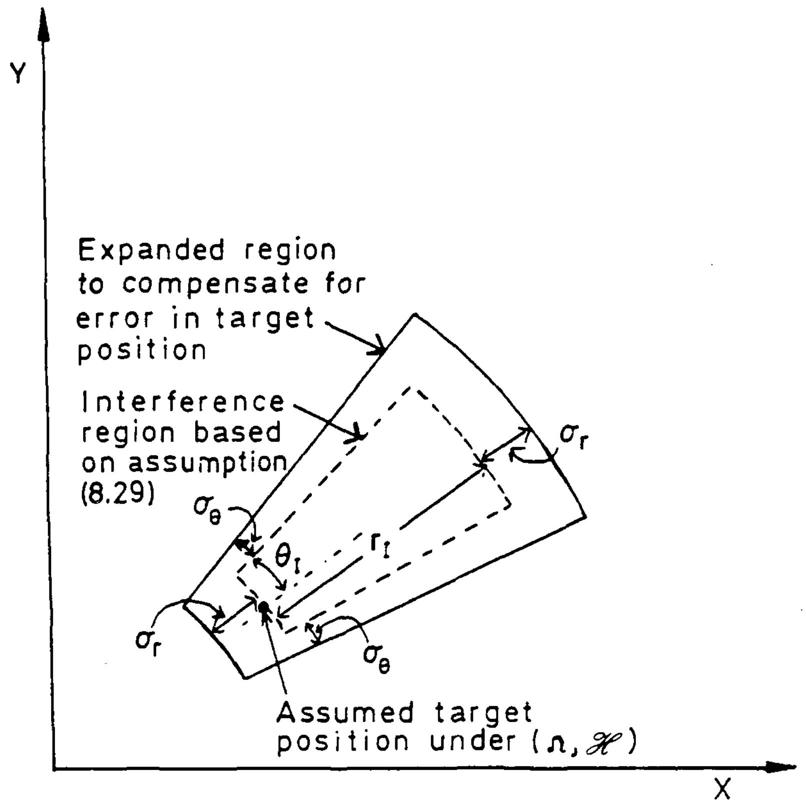


Fig 8.3 Region for evaluation of m_I: heuristic expansion of interference region to compensate for incorrect assumption (8.29)

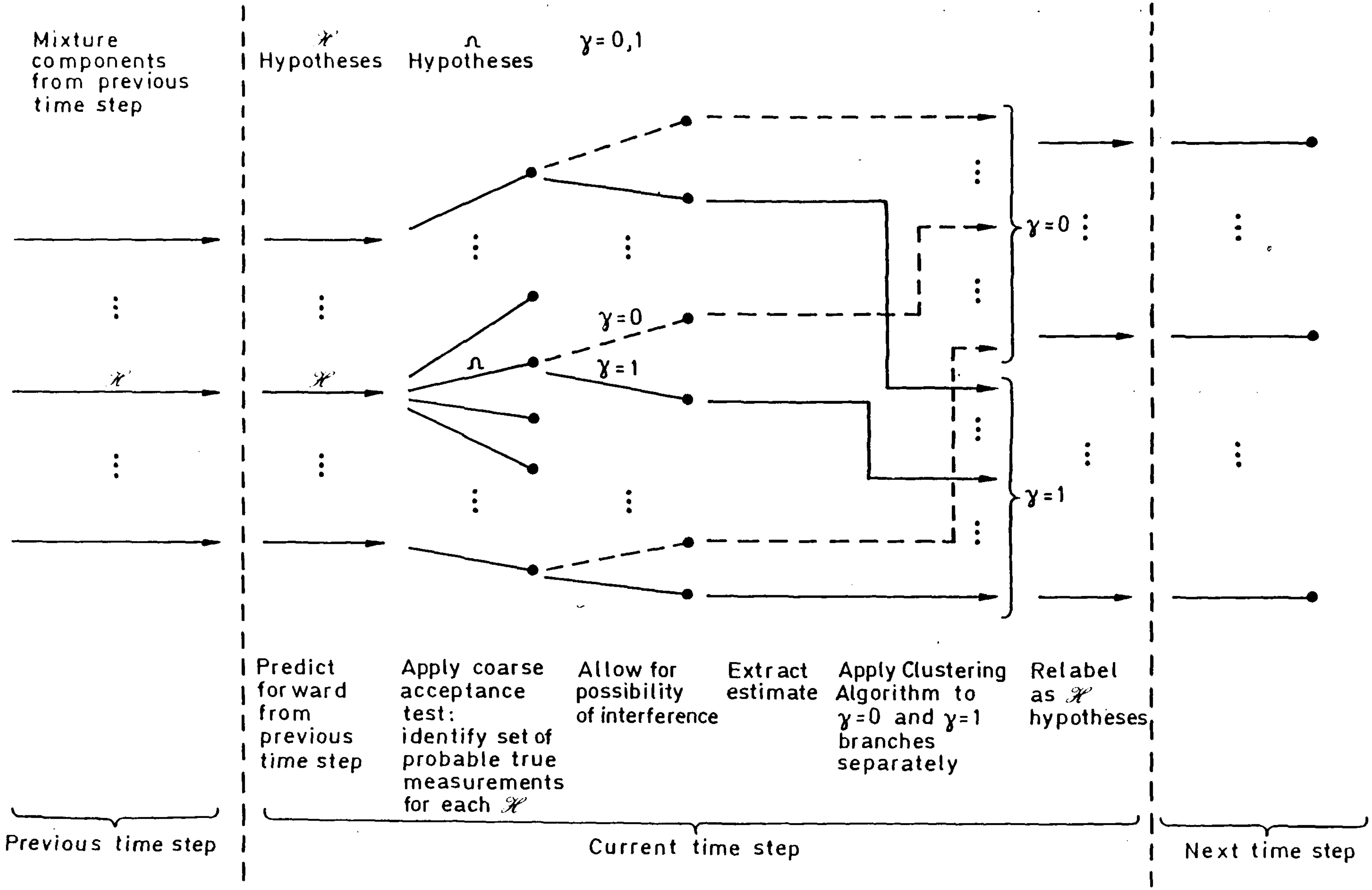


Fig 8.4 Implementation of the interference filter using the Clustering Algorithm

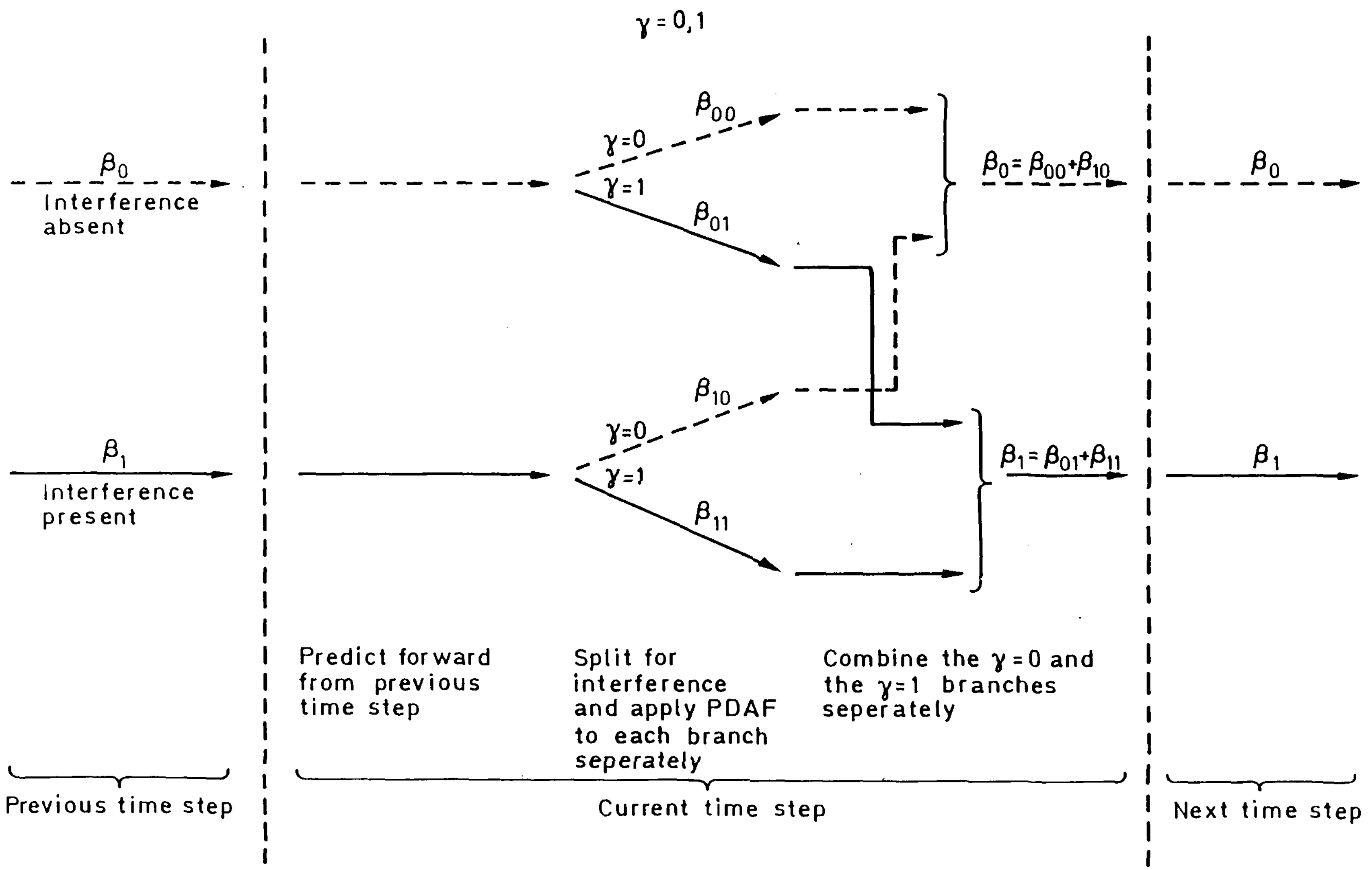


Fig 8.5 Implementation of the interference filter using a PDAF type of approximation

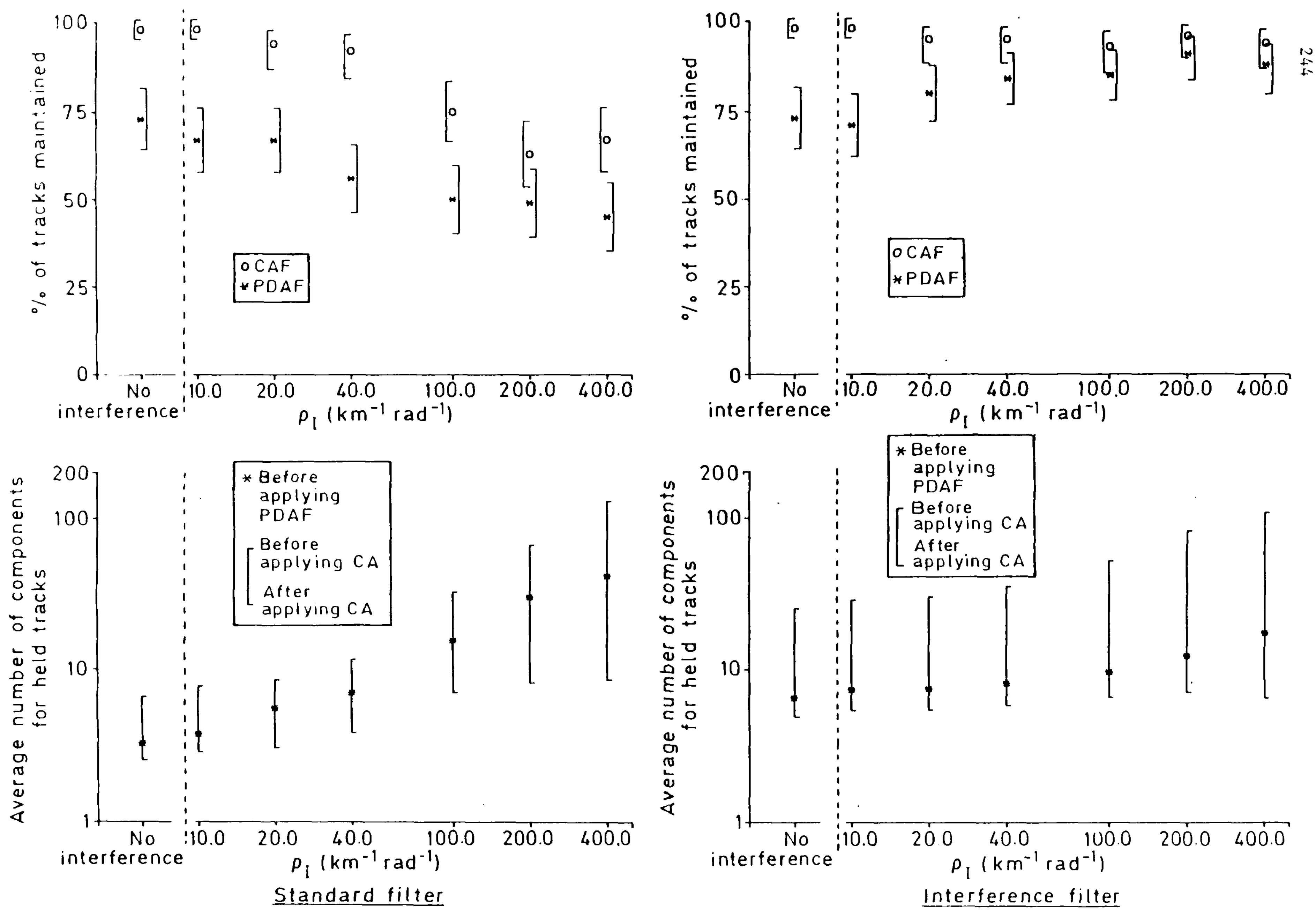


Fig 8.6 Performance of the standard filter and the interference filter for varying ρ_I

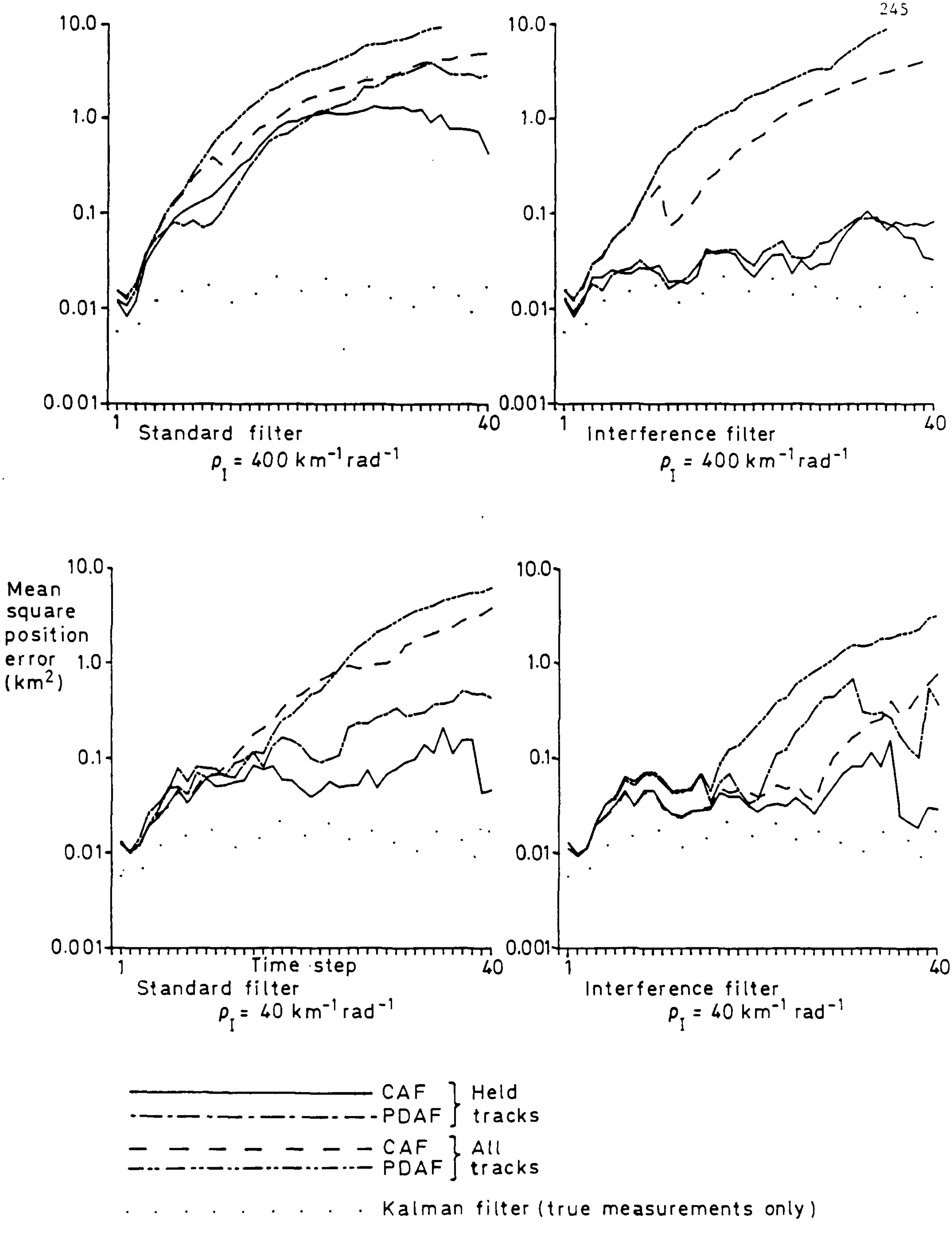


Fig 8.7 Mean square target position error for the intermittent interference problem for ρ_I = 40 and 400 km^{-1} rad^{-1}

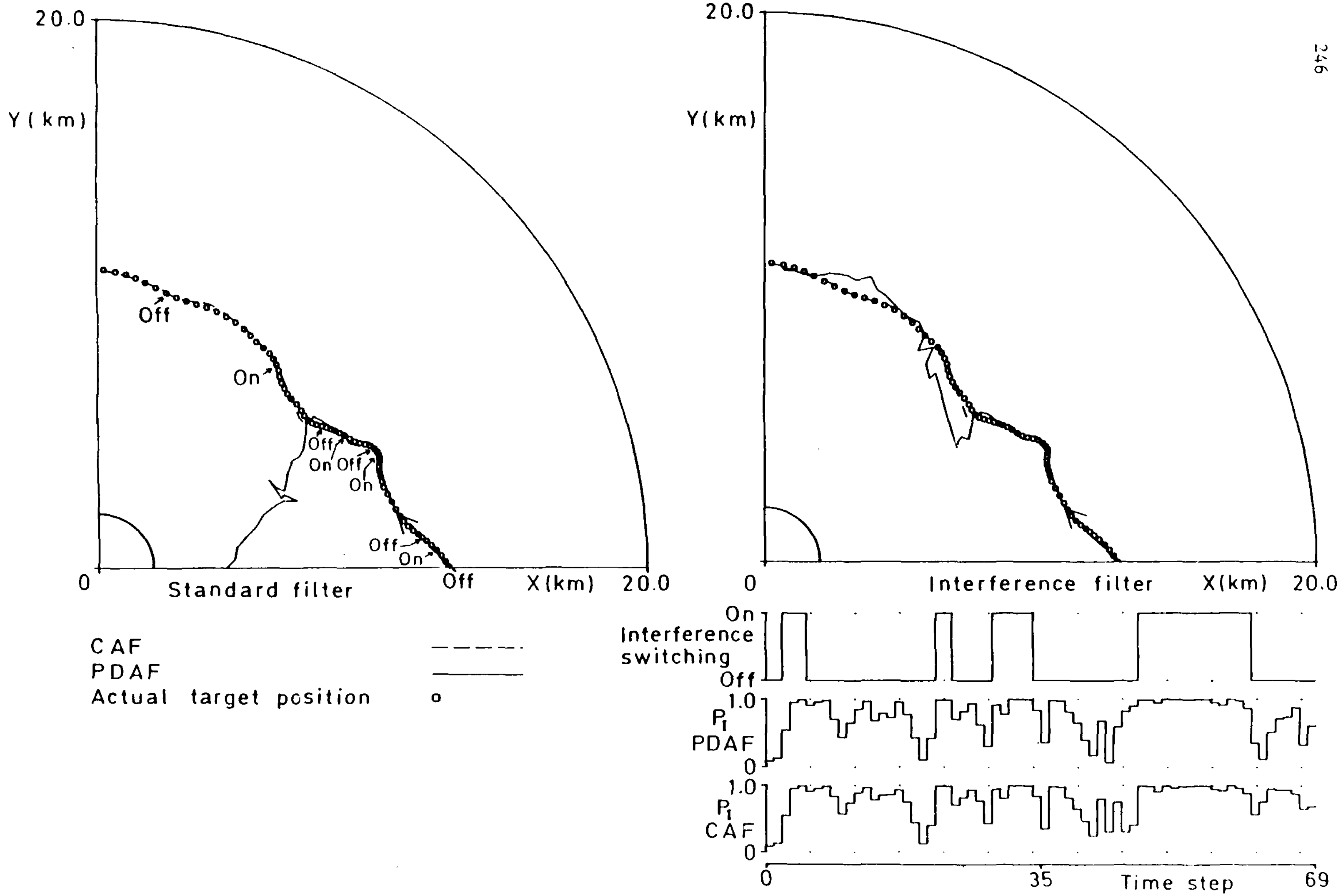


Fig 8.8 An example of tracking with intermittent interference of density $\rho_I = 10\ km^{-1}\ rad^{-1}$

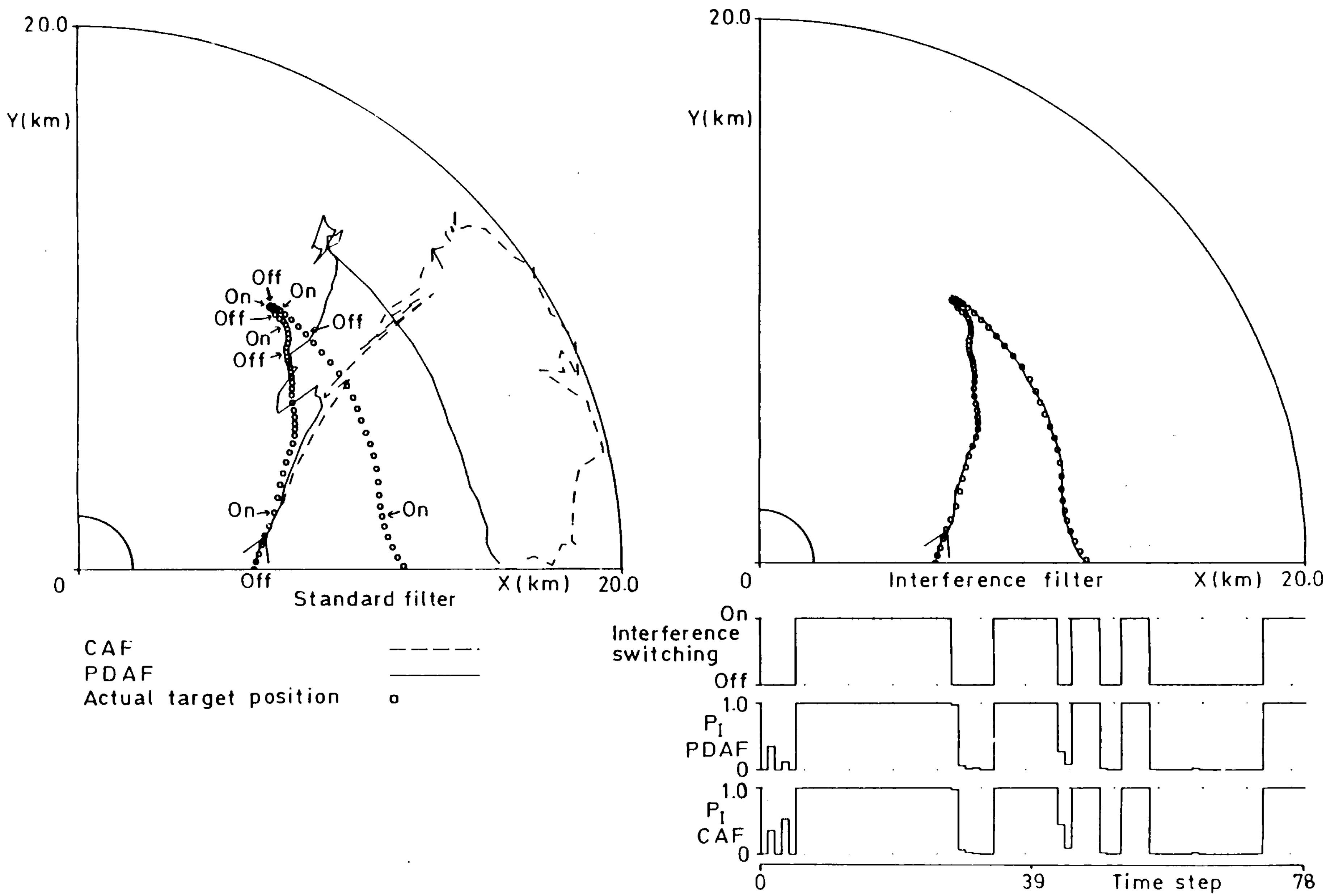


Fig 8.9 An example of tracking with intermittent interference of density $\rho_I = 40\ km^{-1}\ rad^{-1}$

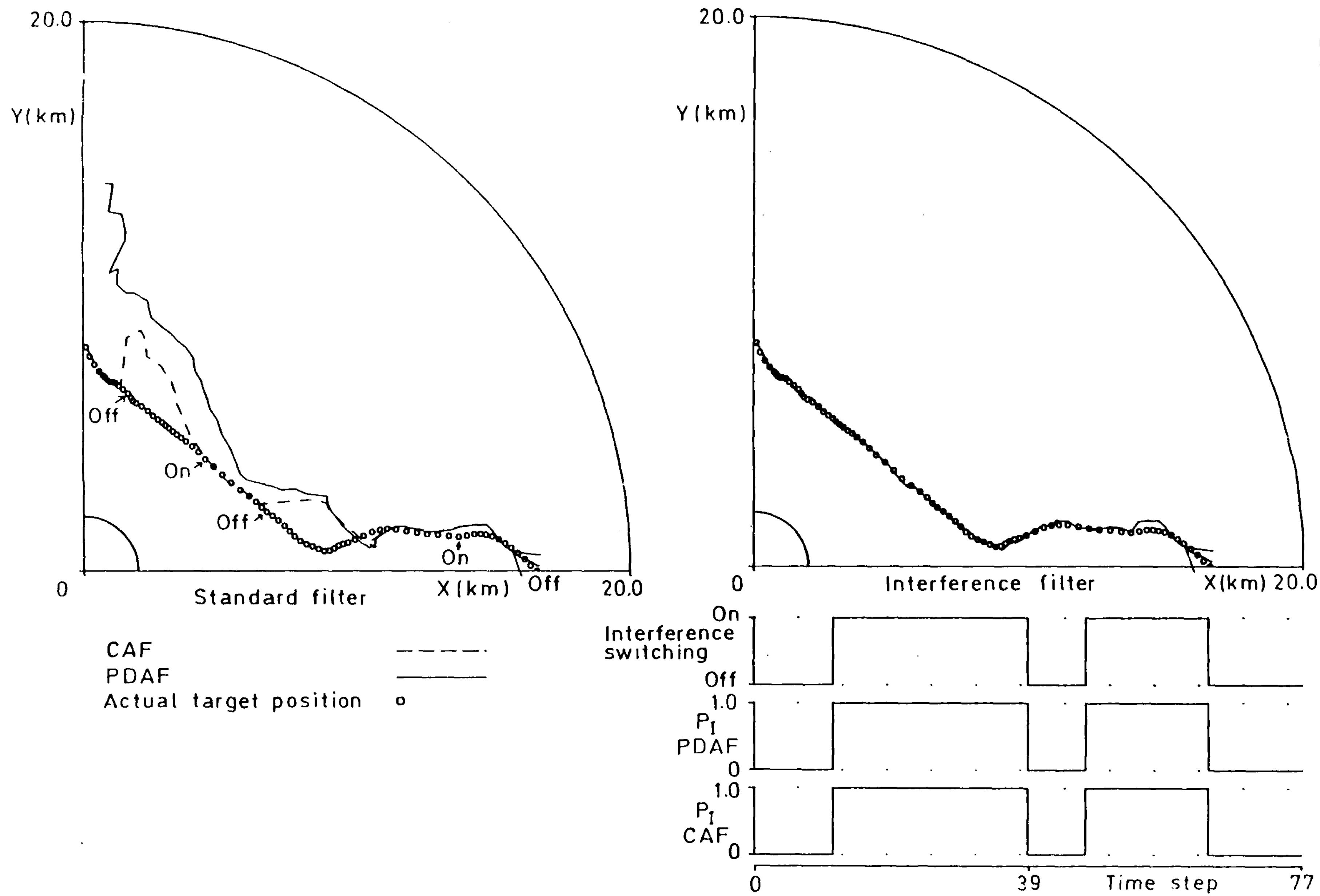


Fig 8.10 An example of tracking with intermittent interference of density $\rho_I = 400$ km^{-1} rad^{-1}

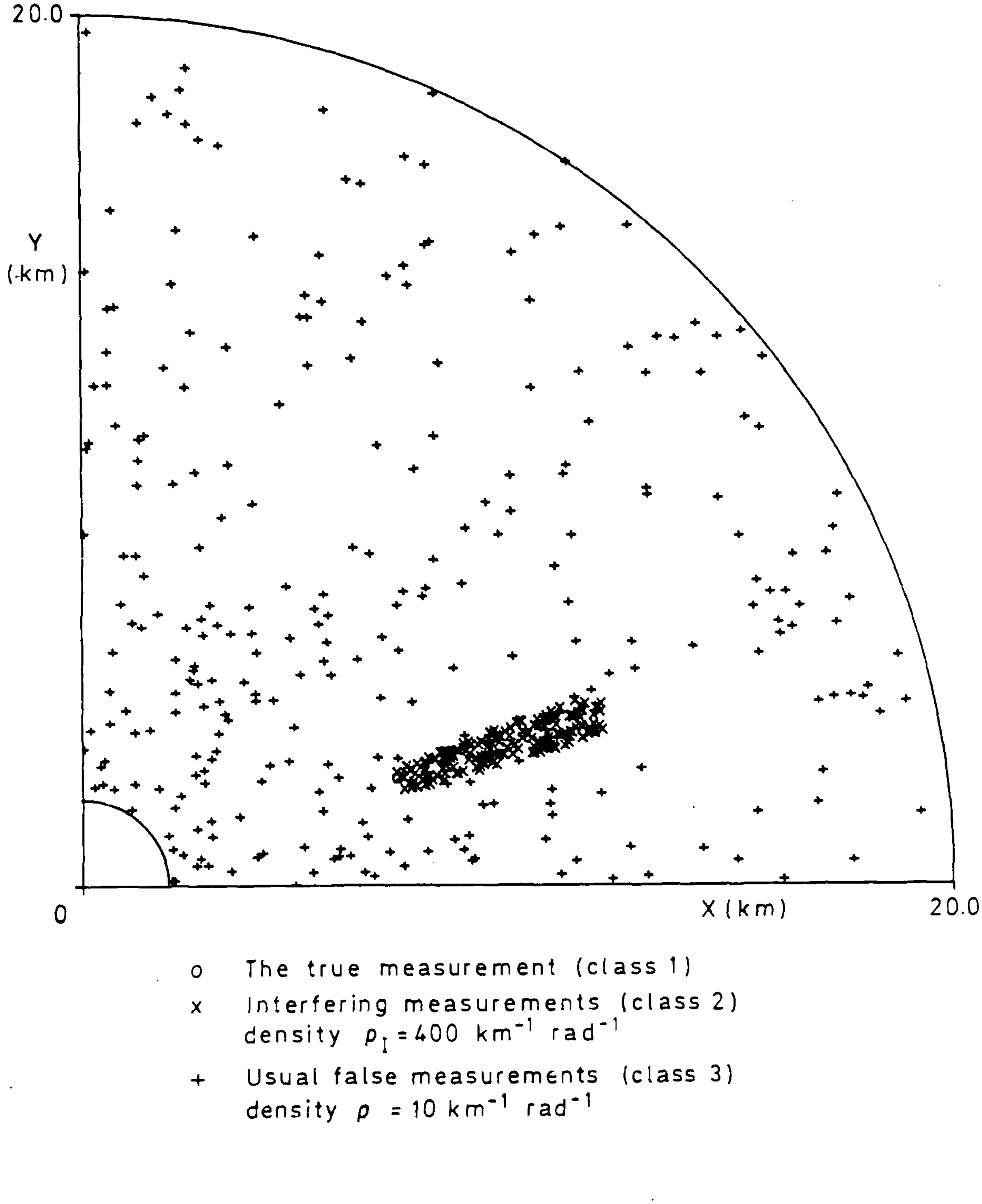


Fig 8.11 Sample plot of all measurements received on a single scan

9 CONCLUSIONS AND FURTHER WORK

In this thesis we have shown how Bayesian techniques may be applied to tracking problems where the origin of the measurement is uncertain. A mixture reduction technique has been developed to contain the ever growing computational requirements of the optimal Bayesian filter. The performance of this Clustering Algorithm has been assessed by simulation for a straightforward baseline tracking problem, and it has been compared with the PDAF method. Filters have also been developed for extensions of the baseline case including data fusion and measurement interference problems.

The detailed conclusions and discussions for this study are given at the end of each chapter. Some overall observations are given below:

(i) The performance of the CAF is always better than or similar to that of the PDAF. This improvement is at the expense of of increased computational memory and processing requirements. The processing time for the CAF is usually within an order of magnitude of the PDAF processing time, although for very difficult cases, where performance is in any case poor, the excess may be several orders of magnitude.

(ii) Bayes theorem provides a convenient recursive mechanism for incorporating information from various sources, and for many interesting tracking problems a filter based on the optimal solution may be derived. However, even for minor extensions of the baseline problem, the optimal filter may be very complex so that a number of significant approximations must be imposed to obtain a practical filter.

(iii) Simulation has proved to be a useful tool, both for performance assessment and as an aid to understanding the operation of the filters.

This study has been concerned with estimating the current state of a single target based on past measurements. We propose to extend this work to include trajectory estimation and multiple target tracking.

For some applications it is necessary to estimate the past trajectory of a target as well as its current position. Each new measurement that is received provides information on the past values of the state vector via the target model, and clearly this information should be used for trajectory estimation. A filter which refines past estimates in the light of subsequent measurements is called a smoothing filter. In terms of the pdf of target state, for a smoothing filter we require:

$$p\left(\underline{x}_k | Z_1, \ldots, Z_n\right) ,$$

where $k \leqslant n$. For standard filtering problems without measurement uncertainty, efficient optimal smoothing algorithms have been derived (see Jazwinski[27]). For trajectory estimation, it has been shown that these filters can provide an impressive improvement over the standard Kalman filter (see Refs 53 and 54). The smoothing problem for uncertain measurement association is more complex. Mahalanabis and Zhou[55] have suggested smoothing back one or two time steps to improve a PDAF estimate. Also we have obtained some encouraging preliminary results for full trajectory estimation using a PDAF based smoothing algorithm. We hope to extend this study to investigate the merrits of retaining more than one component for the smoothing operation.

The problem of tracking multiple targets is more complicated than the single target case. This is due to the range of extra measurement association hypotheses that must be taken into account. The coarse acceptance test is most valuable here in eliminating improbable associations between measurements and remote tracks. Blackman[1] presents a branching algorithm for generating the appropriate hypotheses which is based on techniques developed by Reid[23] and Mori *et al*[26]. As for the single target case, the number of feasible hypotheses grows rapidly, and we intend to investigate the application of the Clustering Algorithm to control this growth. Also we propose to study the multiple target data fusion problem.

Acknowledgments

I should like to thank my supervisors, Prof D.P. Atherton and Prof J.A. Bather of the University of Sussex and Mr R.G. Hollingworth of the Royal Aerospace Establishment, for their guidance and encouragement throughout the course of this work. I should also like to acknowledge the assistance of my colleague Mr C.J. Taylor who wrote most of the computer graphics programs for the illustrations of this thesis.

Financial support for this study was provided by the Royal Aerospace Establishment.

Appendix A

THE KALMAN FILTER RELATIONS

A.1 The Kalman filter problem

The Kalman filter problem is similar to the problem statement of section 2.2 of the main text, except that only a single true measurement is available at each time step. A simple form of the Kalman filter problem is stated below.

The state vector $\underline{x}$ is assumed to obey a linear system model:

$$\underline{x}_{k+1} = \Phi \underline{x}_k + \Gamma \underline{w}_k \quad , \tag{A-1}$$

where $\underline{x}_k$ is the n-dimensional state vector at time t_k ,

Φ is the $n \times n$ *state transition* matrix,

Γ is the $n \times r$ distribution matrix,

and $\underline{w}_k$ is the r-dimensional system driving noise which has a Gaussian distribution with zero mean and covariance given by:

$$E\left[\underline{w}_\ell \, \underline{w}_k^T\right] = Q\delta_{\ell k} \quad .$$

Here Q is a positive definite $r \times r$ matrix and $\delta_{\ell k}$ is the Kronecker delta. At each time step t_k , a u-dimensional measurement vector $\underline{z}_k$ is available, which is linearly related to the state vector:

$$\underline{z}_k = H\underline{x}_k + \underline{v}_k \quad , \tag{A-2}$$

where H is the $u \times n$ measurement matrix

and $\underline{v}_k$ is the u-dimensional measurement noise which has a Gaussian distribution with zero mean and covariance given by:

$$E\left[\underline{v}_\ell \, \underline{v}_k^T\right] = R\delta_{\ell k} \quad .$$

Here R is a positive definite $u \times u$ matrix and $\delta_{\ell k}$ is the Kronecker delta. Also it is assumed that initially at time t_1 , the state vector $\underline{x}_1$ is known to have a Gaussian distribution with mean $\bar{\underline{x}}_1$ and covariance M_1 . In a more general formulation, the covariance matrices and system matrices may depend on k . However the resulting filter is similar and so for simplicity of notation, this dependence is not included.

Using this information, the problem is to determine the pdf of the state vector at each time step t_k , conditional on all the measurements received up to and including t_k . From this pdf an optimal estimate according to any desired criterion may be obtained.

Since all relationships are linear and all distributions Gaussian, the required pdf of the state vector at each time step is also Gaussian (as is shown in what follows). This is why a particularly neat and elegant recursive solution may be obtained. The Kalman filter recursion at each time step is essentially a two stage process. In the first stage, the prior pdf at t_k is updated with the measurement $\underline{z}_k$ to obtain the required posterior pdf. In the second stage, this posterior pdf is predicted forwards to obtain the prior pdf for the following time step t_{k+1} . The recursions for these two stages will be obtained using Bayesian techniques in the following two sections. (Different methods and optimization criteria which also lead to the Kalman filter relations are detailed in Refs 27 to 31.)

A.2 Update of the prior pdf

Suppose that the prior pdf at time t_k (conditioned on all measurements up to and including $\underline{z}_{k-1}$) is given by:

$$p\left(\underline{x}_k \middle| Z_{k-1}\right) = \mathcal{N}\left(\underline{x}_k ; \bar{\underline{x}}_k , M_k\right) ,$$

where the mean $\bar{\underline{x}}_k$ and covariance M_k are known and $Z_\ell = \left\{\underline{z}_1, \ldots, \underline{z}_\ell\right\}$. This is true for $k = 1$ and by induction, it will be established for all k .

The required posterior pdf may be obtained directly from Bayes theorem:

$$p\left(\underline{x}_k \middle| Z_k\right) = \frac{p\left(\underline{z}_k \middle| \underline{x}_k , Z_{k-1}\right) p\left(\underline{x}_k \middle| Z_{k-1}\right)}{p\left(\underline{z}_k \middle| Z_{k-1}\right)} \qquad \text{(A-4)}$$

where $p(\underline{z}_k | Z_{k-1}) = \int p(\underline{z}_k | \underline{x}_k , Z_{k-1})\, p(\underline{x}_k | Z_{k-1})\, d\underline{x}_k$ is the normalizing constant. Now:

$$p\left(\underline{z}_k \middle| \underline{x}_k , Z_{k-1}\right) = p\left(\underline{z}_k \middle| \underline{x}_k\right) ,$$

since given $\underline{x}_k$, Z_{k-1} contributes no extra information and from equation (A-2):

$$p\left(\underline{z}_k \middle| \underline{x}_k\right) = \mathcal{N}\left(\underline{z}_k ; H\underline{x}_k , R\right) .$$

Hence the numerator of equation (A-4) is given by:

$$\mathcal{N}\left(\underline{z}_k ; H\underline{x}_k , R\right)\mathcal{N}\left(\underline{x}_k ; \bar{\underline{x}}_k , M_k\right)$$

$$= \left[(2\pi)^{n+m}|R|\,|M_k|\right]^{-\frac{1}{2}} \exp\left\{- \tfrac{1}{2}\left(\underline{z}_k - H\underline{x}_k\right)^T R^{-1}\left(\underline{z}_k - H\underline{x}_k\right) - \tfrac{1}{2}\left(\underline{x}_k - \bar{\underline{x}}_k\right)^T M_k^{-1}\left(\underline{x}_k - \bar{\underline{x}}_k\right)\right\}$$

$$= \left[(2\pi)^{n+m}|R|\,|M_k|\right]^{-\frac{1}{2}} \exp\left\{- \tfrac{1}{2}\left(\underline{x}_k - \hat{\underline{x}}_k\right)^T P_k^{-1}\left(\underline{x}_k - \hat{\underline{x}}_k\right) + r'\right\} \qquad \text{(A-5)}$$

on combining the quadratic forms (see section A.4), where

$$P_k^{-1} = M_k^{-1} + H^T R^{-1} H \qquad \text{(A-6)}$$

$$\hat{\underline{x}}_k = \bar{\underline{x}}_k + P_k H^T R^{-1}\left(\underline{z}_k - H\bar{\underline{x}}_k\right) \qquad \text{(A-7)}$$

and

$$r' = - \tfrac{1}{2}\left(\underline{z}_k - H\bar{\underline{x}}_k\right)^T\left(HM_kH^T + R\right)^{-1}\left(\underline{z}_k - H\bar{\underline{x}}_k\right)$$

which is independent of $\underline{x}_k$.

The denominator of equation (A-4) is the integral of equation (A-5) with respect to $\underline{x}_k$:

$$p\left(\underline{z}_k \middle| Z_{k-1}\right) = \left[(2\pi)^{m+n}|R|\,|M_k|\right]^{-\frac{1}{2}} e^{r'}\left[(2\pi)^n|P_k|\right]^{\frac{1}{2}} . \qquad \text{(A-8)}$$

Dividing equation (A-5) by equation (A-8) gives:

$$p\left(\underline{x}_k \middle| Z_k\right) = \mathcal{N}\left(\underline{x}_k ; \hat{\underline{x}}_k , P_k\right) . \qquad \text{(A-9)}$$

So the posterior pdf of $\underline{x}$ is Gaussian with mean $\hat{\underline{x}}_k$ and covariance P_k . The expression for P_k may be written in a more convenient form using the matrix inversion theorem (see Ref 27 page 262) to obtain,

with equation (A-7), the update recursions in the forms that are usually quoted for the Kalman filter:

$$\hat{\underline{x}}_k = \bar{\underline{x}}_k + K_k \underline{\nu}_k \tag{A-10}$$

where $\underline{\nu}_k = \underline{z}_k - H\bar{\underline{x}}_k$ is known as the innovation, and $K_k = P_k H^T R^{-1}$ is the Kalman gain. The covariance P_k, which is required for the evaluation of the gain matrix is given by: (from equation (A-6))

$$P_k = M_k - M_k H^T S_k^{-1} H M_k \tag{A-11}$$

where $S_k = H M_k H^T + R$.

We can now verify that S_k is the prior covariance of $\underline{z}_k$. The prior pdf of $\underline{z}_k$ is given by equation (A-8), from which:

$$p\left(\underline{z}_k \middle| Z_{k-1}\right) = \left[(2\pi)^m |R| \, |M_k| \, |P_k^{-1}|\right]^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}\left(\underline{z}_k - H\bar{\underline{x}}_k\right)^T S_k^{-1}\left(\underline{z}_k - H\bar{\underline{x}}_k\right)\right\} .$$

Also

$$|R| \, |M_k| \, |P_k^{-1}| = |R| \, |M_k| \, |M_k^{-1} + H^T R^{-1} H|$$

$$= |H M_k H^T + R| = |S_k| ,$$

using standard identities for determinants (see, for example Sorenson[36]). Hence:

$$p\left(\underline{z}_k \middle| Z_{k-1}\right) = \mathcal{N}\left(\underline{z}_k \, ; H\bar{\underline{x}}_k \, , S_k\right) . \tag{A-12}$$

Note also, that since $\underline{\nu}_k = \underline{z}_k - H\bar{\underline{x}}_k$, the prior distribution of the innovation is Gaussian with zero mean and covariance S_k .

A.3 Prediction

To complete the recursion, it is necessary to predict forwards from the posterior pdf at t_k to obtain the prior pdf at time t_{k+1}, *ie*

$$p(\underline{x}_{k+1}|Z_k) \ .$$

Now, by definition,

$$p(\underline{x}_{k+1}|Z_k) = \int p(\underline{x}_{k+1},\underline{x}_k|Z_k)d\underline{x}_k$$

$$= \int p(\underline{x}_{k+1}|\underline{x}_k,Z_k)p(\underline{x}_k|Z_k)d\underline{x}_k \ . \tag{A-13}$$

$p(\underline{x}_{k+1}|\underline{x}_k,Z_k) = p(\underline{x}_{k+1}|\underline{x}_k)$, since given $\underline{x}_k$, Z_k contributes no useful information, and from (A-1)

$$p(\underline{x}_{k+1}|\underline{x}_k) = \mathcal{N}(\underline{x}_{k+1}|\Phi\underline{x}_k,\Gamma Q\Gamma^T) \ . \tag{A-14}$$

Hence, from (A-9) and (A-14)

$$\begin{aligned} p(\underline{x}_{k+1},\underline{x}_k|Z_k) &= \left[(2\pi)^{2n}|\Gamma Q\Gamma^T||P_k|\right]^{-\frac{1}{2}} \exp\Big\{-\tfrac{1}{2}(\underline{x}_{k+1} - \Phi\underline{x}_k)^T(\Gamma Q\Gamma^T)^{-1}(\underline{x}_{k+1} - \Phi\underline{x}_k) \\ &\qquad - \tfrac{1}{2}(\underline{x}_k - \hat{\underline{x}}_k)^T P_k^{-1}(\underline{x}_k - \hat{\underline{x}}_k)\Big\} \\ &= \left[(2\pi)^{2n}|\Gamma Q\Gamma^T||P_k|\right]^{-\frac{1}{2}} \exp\Big\{-\tfrac{1}{2}(\underline{x}_k - \underline{d})^T D^{-1}(\underline{x}_k - \underline{d}) \\ &\qquad - \tfrac{1}{2}(\underline{x}_{k+1} - \Phi\hat{\underline{x}}_k)^T M_{k+1}^{-1}(\underline{x}_{k+1} - \Phi\hat{\underline{x}}_k)\Big\} \end{aligned}$$

......(A-15)

on rearranging the quadratic forms (using the result of section A.4), where $\underline{d}$ is independent of $\underline{x}_k$,

$$D^{-1} = \Phi^T(\Gamma Q\Gamma^T)^{-1}\Phi + P_k^{-1}$$

and $$M_{k+1} = \Phi P_k\Phi^T + \Gamma Q\Gamma^T \ .$$

The expression (A-15) may be integrated with respect to $\underline{x}$ to give

$$\int p(\underline{x}_{k+1},\underline{x}_k|Z_k)d\underline{x}_k$$

$$= |D|^{\frac{1}{2}}\left[(2\pi)^n|\Gamma Q\Gamma^T||P|\right]^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(\underline{x}_{k+1} - \Phi\hat{\underline{x}}_k)^T M_{k+1}^{-1}(\underline{x}_{k+1} - \Phi\hat{\underline{x}}_k)\right\} .$$

Also

$$\begin{aligned}|D|^{\frac{1}{2}}(|\Gamma Q\Gamma^T||P_k|)^{-\frac{1}{2}} &= (|D^{-1}||\Gamma Q\Gamma^T||P_k|^{-\frac{1}{2}} \\ &= (|P_k^{-1}||\Gamma Q\Gamma^T + \Phi P_k\Phi^T||P_k|)^{-\frac{1}{2}} \\ &= |M_k|^{-\frac{1}{2}}\end{aligned}$$

using standard identities for determinants (see Ref 36).

Therefore,

$$p(\underline{x}_{k+1}|Z_k) = \mathcal{N}(\underline{x}_{k+1};\bar{\underline{x}}_{k+1},M_{k+1}) , \qquad \text{(A-16)}$$

where $\bar{\underline{x}}_{k+1} = \Phi\hat{\underline{x}}$

and $M_{k+1} = \Phi P_k\Phi^T + \Gamma Q\Gamma^T$.

These expressions for the mean $\bar{\underline{x}}_{k+1}$ and covariance M_{k+1} , complete the Kalman filter recursions.

A.4 Combination of quadratic forms

Lemma If B and C are symmetric and positive definite, then

$$(\underline{a} - A\underline{x})^T B^{-1}(\underline{a} - A\underline{x}) + (\underline{b} - \underline{x})^T C^{-1}(\underline{b} - \underline{x}) = (\underline{x} - \underline{y})^T D^{-1}(\underline{x} - \underline{y}) + r' , \qquad \text{(A-17)}$$

where $\underline{y} = \underline{b} + DA^TB^{-1}(\underline{a} - A\underline{b})$,

$D^{-1} = A^TB^{-1}A + C^{-1}$

and $r' = (\underline{a} - A\underline{b})^T(B + ACA^T)^{-1}(\underline{a} - A\underline{b})$.

Note that r' does not depend on $\underline{x}$.

<u>Proof</u> Consider the left-hand side of (A-17):

$$
\begin{aligned}
(\underline{a} - A\underline{x})^T B^{-1}(\underline{a} - A\underline{x}) + (\underline{b} - \underline{x})^T C^{-1}(\underline{b} - \underline{x}) &= \underline{x}^T(A^T B^{-1} A + C^{-1})\underline{x} \\
&\quad - \underline{x}^T(A^T B^{-1}\underline{a} + C^{-1}\underline{b}) - (\underline{a}^T B^{-1} A + \underline{b}^T C^{-1})\underline{x} \\
&\quad + \underline{a}^T B^{-1}\underline{a} + \underline{b}^T C^{-1}\underline{b} \\
&= \underline{x}^T D^{-1}\underline{x} \\
&\quad - \underline{x}^T(A^T B^{-1}(\underline{a} - A\underline{b}) + D^{-1}\underline{b}) \\
&\quad - \left((\underline{a} - A\underline{b})^T B^{-1} A + \underline{b}^T D^{-1}\right)\underline{x} \\
&\quad + \underline{a}^T B^{-1}\underline{a} + \underline{b}^T C^{-1}\underline{b} \\
&= \underline{x}^T D^{-1}\underline{x} - \underline{x}^T D^{-1}\underline{y} - y^T D^{-1}\underline{x} \\
&\quad + \underline{a}^T B^{-1}\underline{a} + \underline{b}^T C^{-1}\underline{b} \\
&= (\underline{x} - \underline{y})^T D^{-1}(\underline{x} - \underline{y}) + r' \quad ,
\end{aligned}
$$

where $r' = - \underline{y}^T D^{-1}\underline{y} + \underline{a}^T B^{-1}\underline{a} + \underline{b}^T C^{-1}\underline{b}$

$$
\begin{aligned}
&= - (\underline{a} - A\underline{b})^T B^{-1} A D A^T B^{-1}(\underline{a} - A\underline{b}) - \underline{a}^T B^{-1} A\underline{b} + \underline{b}^T A^T B^{-1} A\underline{b} \\
&\quad - \underline{b}^T A^T B^{-1}\underline{a} + \underline{b}^T A^T B^{-1} A\underline{b} - \underline{b}^T(A^T B^{-1} A + C^{-1})\underline{b} + \underline{a}^T B^{-1}\underline{a} + \underline{b}^T C^{-1}\underline{b} \\
&= (\underline{a} - A\underline{b})^T \left[B^{-1} - B^{-1} A(A^T B^{-1} A + C^{-1})^{-1} A^T B^{-1}\right] (\underline{a} - A\underline{b})
\end{aligned}
$$

and from the matrix inversion lemma (see Ref 27, p 262), the term in square brackets is equal to

$$(B + ACA^T)^{-1}$$

which completes the proof.

Appendix B

MEAN AND COVARIANCE OF A MIXTURE DISTRIBUTION AND THE PDAF ALGORITHM

Consider any mixture distribution with pdf

$$p(\underline{x}) = \sum_{i=1}^{N} \beta_i \, p_i(\underline{x})$$

where $p_i(\underline{x})$ is a component pdf

and β_i is a probability associated with the ith component

such that:

$$\beta_i > 0$$

and

$$\sum_{i=1}^{N} \beta_i = 1 \; .$$

Also let the mean of the ith component be $\hat{\underline{x}}_i$ and let the covariance of the ith component be P_i .

The mean of the mixture is defined by:

$$\hat{\underline{x}} = \int \underline{x} p(\underline{x}) d\underline{x}$$

$$= \sum_{i=1}^{N} \beta_i \int \underline{x} p_i(\underline{x}) d\underline{x} = \sum_{i=1}^{N} \beta_i \, \hat{\underline{x}}_i \; . \qquad \text{(B-1)}$$

The covariance of the mixture is defined by:

$$P = \int (\underline{x} - \hat{\underline{x}})(\underline{x} - \hat{\underline{x}})^T \, p(\underline{x}) \, d\underline{x}$$

$$= \int \underline{x} \, \underline{x}^T \, p(\underline{x}) \, d\underline{x} - \hat{\underline{x}} \, \hat{\underline{x}}^T$$

$$= \sum_{i=1}^{N} \beta_i \int \underline{x} \, \underline{x}^T \, p_i(\underline{x}) \, d\underline{x} - \hat{\underline{x}} \, \hat{\underline{x}}^T \quad .$$

But

$$P_i = \int \underline{x} \, \underline{x}^T \, p_i(\underline{x}) \, d\underline{x} - \hat{\underline{x}}_i \, \hat{\underline{x}}_i^T \quad ,$$

so

$$P = \sum_{i=1}^{N} \beta_i \left(P_i + \hat{\underline{x}}_i \, \hat{\underline{x}}_i^T \right) - \hat{\underline{x}} \, \hat{\underline{x}}^T \quad . \tag{B-2}$$

Another form for this covariance may be obtained by observing that:

$$\sum_{i=1}^{N} \beta_i \, \hat{\underline{x}}_i \, \hat{\underline{x}}_i^T - \hat{\underline{x}} \, \hat{\underline{x}}^T = \sum \beta_i \, \hat{\underline{x}}_i \, \hat{\underline{x}}_i^T - \hat{\underline{x}} \sum \beta_i \, \hat{\underline{x}}_i^T - \sum \beta_i \, \hat{\underline{x}}_i \, \hat{\underline{x}}^T + \hat{\underline{x}} \, \hat{\underline{x}}^T$$

$$= \sum_{i=1}^{N} \beta_i \left(\hat{\underline{x}}_i - \hat{\underline{x}} \right) \left(\hat{\underline{x}}_i - \hat{\underline{x}} \right)^T \quad .$$

Therefore
$$P = \sum_{i=1}^{N} \beta_i \, P_i + \sum_{i=1}^{N} \beta_i \left(\hat{\underline{x}}_i - \hat{\underline{x}} \right) \left(\hat{\underline{x}}_i - \hat{\underline{x}} \right)^T \quad . \tag{B-3}$$

B.2 For the PDAF, the posterior mixture distribution is approximated by a single Gaussian at every time step. The Gaussian approximation is chosen to have the same mean and covariance as the mixture. The PDAF is an efficient algorithm because for this approximation, explicit calculation of the mean and covariance of each individual mixture component may be avoided.

Due to the single Gaussian approximation, the prior pdf at any time step is approximated by:

$$p(\underline{x}|\mathscr{P}) = \mathcal{N}(\underline{x} ; \bar{\underline{x}} , M) \quad .$$

After update by a set Z of m measurements, the posterior pdf is:

$$p(\underline{x}|Z) = \sum_{\ell=0}^{m} \beta_{\ell}\mathcal{N}\left(\underline{x} ; \hat{\underline{x}}_{\ell} , P'_{\ell}\right) \quad , \qquad \text{(B-4)}$$

$$\text{where} \quad \hat{\underline{x}}_{\ell} = \begin{cases} \bar{\underline{x}} + K\underline{\nu}_{\ell} & \text{if} \quad \ell \neq 0 \\ \bar{\underline{x}} & \text{if} \quad \ell = 0 \end{cases}$$

$$\underline{\nu}_{\ell} = \underline{z}_{\ell} - H\bar{\underline{x}}$$

and

$$P'_{\ell} = \begin{cases} P' & \text{if} \quad \ell \neq 0 \\ M & \text{if} \quad \ell = 0 \quad , \end{cases}$$

where P' and K are obtained from the usual Kalman filter update relations, equation (2.8). For the PDAF approximation we only require the mean $\hat{\underline{x}}$ and covariance P of equation (B-4). From equation (B-1), the mean is given by:

$$\hat{\underline{x}} = \sum_{\ell=0}^{m} \beta_{\ell}\, \hat{\underline{x}}_{\ell} = \bar{\underline{x}} + K\underline{\nu} \quad , \qquad \text{(B-5)}$$

$$\text{where} \quad \underline{\nu} = \sum_{\ell=1}^{m} \beta_{\ell}\underline{\nu}_{\ell} \quad .$$

Therefore we have:

$$\underline{\hat{x}}_\ell - \underline{\hat{x}} = \begin{cases} K\left(\underline{\nu}_\ell - \underline{\nu}\right) & \text{if} \quad \ell \neq 0 \\ - K\underline{\nu} & \text{if} \quad \ell = 0 \ . \end{cases}$$

Substituting this into equation (B-3) gives the required covariance P of (B-4):

$$P = \beta_0 M + \sum_{\ell=1}^{m} \beta_\ell P'$$

$$+ K\left[\beta_0 \underline{\nu}\underline{\nu}^T + \sum_{\ell=1}^{m} \beta_\ell\left(\underline{\nu}_\ell \underline{\nu}_\ell^T + \underline{\nu}\underline{\nu}^T\right) - 2\underline{\nu}\underline{\nu}^T\right]K^T$$

$$= \beta_0 M + \left(1 - \beta_0\right)P' + K\left[\sum_{\ell=1}^{m} \beta_\ell \underline{\nu}_\ell \underline{\nu}_\ell^T - \underline{\nu}\underline{\nu}^T\right]K^T \ . \qquad \text{(B-6)}$$

Note that the computational effort necessary to evaluate equations (B-5) and (B-6) is modest in comparison with the full Bayesian filter (see Ref 11).

Appendix C

THE JOINING ALGORITHM WITH MEASURE d_{ij} IS NOT SUBJECT TO REVERSALS

Suppose that at some stage during mixture reduction, the closest components according to the distance measure given by (3.8) of section 3.6.1 have means $\underline{x}$ and $\underline{y}$ and weights β_x and β_y. The distance between these components is d_{min}, where:

$$d^2_{min} = f\left(\beta_x , \beta_y\right) ||\underline{x} - \underline{y}||^2$$

where $||\underline{x} - \underline{y}||^2 = (\underline{x} - \underline{y})^T P^{-1}(\underline{x} - \underline{y})$,

and $f(\beta_x , \beta_y) = \beta_x \beta_y / (\beta_x + \beta_y)$.

As they are closest, these two components are merged to produce a new component with mean:

$$\underline{w} = \frac{\beta_x \underline{x} + \beta_y \underline{y}}{\beta_x + \beta_y}$$

and weight $\beta_w = \beta_x + \beta_y$.

Now consider any other component with mean $\underline{z}$ and weight β_z. The distance between this component and either of the two which have been merged must be greater than or equal to d_{min}, so:

$$d^2_{min} \leqslant d^2_{xz} = f\left(\beta_x , \beta_z\right) ||\underline{x} - \underline{z}||^2 , \qquad \text{(C-1)}$$

and

$$d^2_{min} \leqslant d^2_{yz} = f\left(\beta_y , \beta_z\right) ||\underline{y} - \underline{z}||^2 . \qquad \text{(C-2)}$$

To confirm that the minimum distance increases monotonically as reduction proceeds (*ie* it is not subject to reversals), we must prove that:

$$d_{zw}^2 \geq d_{min}^2 \quad .$$

Now:

$$d_{zw}^2 = f\left(\beta_w , \beta_z\right)||\underline{z} - \underline{w}||^2$$

$$= f\left(\beta_w , \beta_z\right)||\underline{z} - \frac{\beta_x \underline{x} + \beta_y \underline{y}}{\beta_w}||^2$$

$$= f\left(\beta_w , \beta_z\right)||\left(\underline{z} - \underline{y}\right) - \frac{\beta_x}{\beta_w}\left(\underline{x} - \underline{y}\right)||^2$$

$$= f\left(\beta_w , \beta_z\right)\Bigg\{||\underline{z} - \underline{y}||^2 + \frac{\beta_x^2}{\beta_w^2}||\underline{x} - \underline{y}||^2$$

$$+ \frac{\beta_x}{\beta_w}\left[||\underline{z} - \underline{x}||^2 - ||\underline{z} - \underline{y}||^2 - ||\underline{x} - \underline{y}||^2\right]\Bigg\}$$

$$= \frac{f\left(\beta_w , \beta_z\right)}{\beta_w}\left\{\beta_y||\underline{z} - \underline{y}||^2 + \beta_x||\underline{z} - \underline{x}||^2 - \frac{\beta_x \beta_y}{\beta_w}||\underline{x} - \underline{y}||^2\right\} \quad .$$

Since

$$\frac{f\left(\beta_w , \beta_z\right)}{\beta_w} = \frac{\beta_w \beta_z}{\beta_w\left(\beta_w + \beta_z\right)} = \frac{\beta_z}{\beta_w + \beta_z}$$

and using the definition of the distance measure:

$$d_{zw}^2 = \frac{1}{\beta_w + \beta_z}\left\{\left(\beta_y + \beta_z\right)d_{yz}^2 + \left(\beta_x + \beta_z\right)d_{xz}^2 - \beta_z\, d_{min}^2\right\} \quad .$$

Hence from (C-1) and (C-2):

$$d_{zw}^2 \geq \frac{1}{\beta_w + \beta_z} \left\{ \left(\beta_y + \beta_z\right) d_{min}^2 + \left(\beta_x + \beta_z\right) d_{min}^2 - \beta_z \, d_{min}^2 \right\}$$

$$= \frac{1}{\beta_w + \beta_z} \left(\beta_y + \beta_z + \beta_x\right) d_{min}^2 = d_{min}^2 \; .$$

This completes the proof.

Appendix D

COMPUTATIONAL REQUIREMENTS OF THE REDUCTION ALGORITHMS

D.1 The Joining Algorithm (Fig 3.1)

As explained in section 3.6.1 the operation of the Joining Algorithm is centred around a symmetric distance matrix (d_{ij}^2) with $d_{ii} = 0$. Thus it is necessary to store the upper triangular part of the matrix which occupies $\frac{N^2 - N}{2}$ storage locations, where N is the original number of components in the mixture.

The most time-consuming operations are the evaluation and comparison of the distance measures. The calculation of each distance involves the evaluation of a quadratic form which requires of the order of n^2 multiplications and n^2 additions, where n is the dimension of the state space. Note however that the matrix P in the distance formula equation (3.8) is constant, since the merging of components preserves the overall mixture covariance. Thus only one matrix inversion suffices for all distance evaluations.

To reduce a mixture from N to M components, the number of distance calculations required is

$$N_{DJ} = \frac{N^2 - N}{2} + \sum_{i=2}^{N-M+1} (N - i)$$

$$= N(N - 2) - \frac{M}{2}(M - 3) \quad . \tag{D-1}$$

The identification of

$$\min_{i,j} \; d_{ij}$$

where $i < j$, requires

$$\frac{m}{2}(m - 1) - 1$$

comparisons at each iteration, where m is the number of remaining components. Thus the total number of comparisons required during the reduction of a mixture from N to M components is

$$N_{CJ} = \sum_{m=M}^{N} \left\{ \frac{m}{2}(m - 1) - 1 \right\}$$

$$= \frac{1}{6}(N - M + 1)\left\{ N^2 + N(M - 1) + (M - 1)^2 - 7 \right\} \quad . \quad \text{(D-2)}$$

There are $N - M + 1$ terms in the summation because one extra evaluation of $\min_{i,j} d_{ij}$ is required for the algorithm stopping criterion. Note that the required number of comparisons is of order N^3 and the number of distance calculations is of order N^2. The number of these operations is shown in Fig D.1 as a function of M, for the cases $N = 100$ and $N = 15$. The value of N clearly dominates the number of operations, and although this decreases with M, the decrease is small while $M < \frac{N}{2}$. Note that the number of comparisons required to find the components with the lowest β weights (see Fig 3.1) has not been included in the above total as their number is relatively insignificant.

D.2 The Clustering Algorithm (Fig 3.12)

Unlike the Joining Algorithm whose computational cost can be predicted quite accurately, the cost of the Clustering Algorithm is very dependent on how quickly the mixture components are clustered and on how many iterations are required to adequately reduce the mixture. The most time-consuming operations for the algorithm are distance evaluations and comparisons; the merging of selected components into a single Gaussian is relatively inexpensive.

In the formation of a single cluster, the distance from the cluster centre to every unclustered component must be evaluated. The total number of distance calculations required for an iteration of the algorithm is

$$N'_{DC} = M'(N' - 1) - \sum_{i=1}^{M'-1} (M' - i)m_i \tag{D-3}$$

where N' is the number of components at the start of the iteration, M' is the total number of clusters formed during the iteration and m_i is the number of components combined into the ith cluster. For given N' and M', bounds on N'_{DC} may be obtained by considering the most and least favourable values for m_i. The lower bound is obtained when $N' - (M' - 1)$ components are combined into the first cluster so that all further clusters only contain one element, *ie*

$$m_i = \begin{cases} N' - (M' - 1) & \text{if} \quad i = 1 \\ 1 & \text{otherwise} \end{cases} \quad . \tag{D-4}$$

Thus the lower bound on N'_{DC} is given by, from equation (D-3),

$$L'_{DC} = N' + \frac{M'}{2}(M' - 3) \quad . \tag{D-5}$$

The upper bound is obtained if the first $M' - 1$ clusters only contain one component, so

$$m_i = 1 \qquad \text{for} \qquad i \leqslant M' - 1$$

and

$$1 \leqslant m_{M'} \leqslant N' - (M' - 1) \quad . \tag{D-6}$$

Thus the upper bound on N'_{DC} is given by, from equation (D-3),

$$U'_{DC} = M'\left(N' - \frac{1}{2}(M'+1)\right) \quad . \tag{D-7}$$

Also, since the distance measure for each cluster is normalized by the covariance of the cluster centre (see equation (3.9)), M' matrix inversions are required. In Fig D.2, U'_{DC} and L'_{DC} are shown as a function of M' for $N' = 100$ and $N' = 15$.

To select the components for clustering, each of the N'_{DC} distances must be compared with the clustering theshold. Also comparisons are required to identify those components which are closest together so that they can be merged if no components are clustered. However if components are clustered the minimum distance is no longer required, and so the search for the closest components is abandoned at this stage. Thus the minimum number L'_{CC} of comparisons required for an iteration occurs when m_i is given by equation (D-4) and when the first component to be examined is clustered, so

$$L'_{CC} = L'_{DC} \quad . \tag{D-8}$$

The maximum number U'_{CC} of comparisons required for an iteration occurs when every cluster contains only one component (no components are clustered), *ie*

$$m_i = 1 \qquad \text{for all } i \; .$$

In this case

$$U'_{CC} = U'_{DC} + \left(U'_{DC} - 1\right) \quad . \tag{D-9}$$

Clearly for mixtures with a large number of components, such as $N' = 100$, the first iteration of the Clustering Algorithm could involve a very large number of distance evaluations and comparisons (see Fig D.2). However, in practice it has been found that the number of operations is usually well below the upper bounds U'_{DC} and U'_{CC} , and that mixtures

with a large number of components are usually significantly reduced after the first iteration (*ie* $M' \ll N'$). Thus if further iterations are necessary, the number of components involved is usually fairly modest.

The Clustering Algorithm would be most expensive in the unlikely circumstance of no component ever being clustered. In this case the closest two components would be combined at each iteration, so the mixture would only be reduced by one component per iteration. This provides an upper bound on the total number of operations. For this worst case we also assume that $B_T = 0$, so that every one of the N' components at the start of an iteration is considered as a possible cluster centre. Thus the number of distance evaluatiors and comparisons for each iteration is given by U'_{DC} and U'_{CC} with $M' = N'$. Also N' matrix inversions are required for each iteration. Thus an upper bound on the total number of matrix inversions required to reduce a mixture from N to M components is given by

$$\sum_{N'=M+1}^{N} N' = \frac{1}{2}(N - M)(N + M + 1) \quad . \tag{D-10}$$

The total number of distance calculations is bounded by

$$U_{DC} = \sum_{N'=M+1}^{N} \frac{1}{2}N'(N' - 1) = \frac{1}{6}(N - M)(N^2 + NM + M^2 - 1)$$

..... (D-11)

and the total number of comparisons is bounded by

$$U_{CC} = 2U_{DC} - (N - M) \quad . \tag{D-12}$$

These upper bounds on the number of operations are shown in Fig D.3 as a function of M, for $N = 100$ and $N = 15$. In the best possible case all components that are clustered are combined into the first cluster on the first iteration. Thus M is a lower bound on the total number of matrix inversions and equation (D-5), with $N' = N$ and $M' = M$, gives a lower bound on the total number of distance calculations and comparisons.

D.3 Comparison of operation counts for the two algorithms

If the original number N of components in the mixture is large compared with the number M of components after reduction, the number of operations required by the Joining Algorithm lies between the upper and lower bounds of the number of operations for the Clustering Algorithm. This is shown in Table D.1. For the simulation example reported in Chapter 4, the Clustering Algorithm was consistently more efficient than the Joining Algorithm. Also it should be noted that for the Joining Algorithm a large distance matrix must be stored. For the Clustering Algorithm, storage requirements over those necessary to hold the mixture components are negligible.

Table D.1

Operation counts for the Joining Algorithm and the Clustering Algorithm when N is large compared with M

Operation	Joining Algorithm	Upper bound for Clustering Algorithm	Lower bound for Clustering Algorithm
Distance calculations	$N_{DJ} = O(N^2)$	$U_{DC} = O(N^3)$	$L_{DC} = O(N)$
Comparisons	$N_{CJ} = O(N^3)$	$U_{CC} \simeq 2N_{CJ}$	$L_{CC} = O(N)$
Matrix inversions	1	$O(N^2)$	M

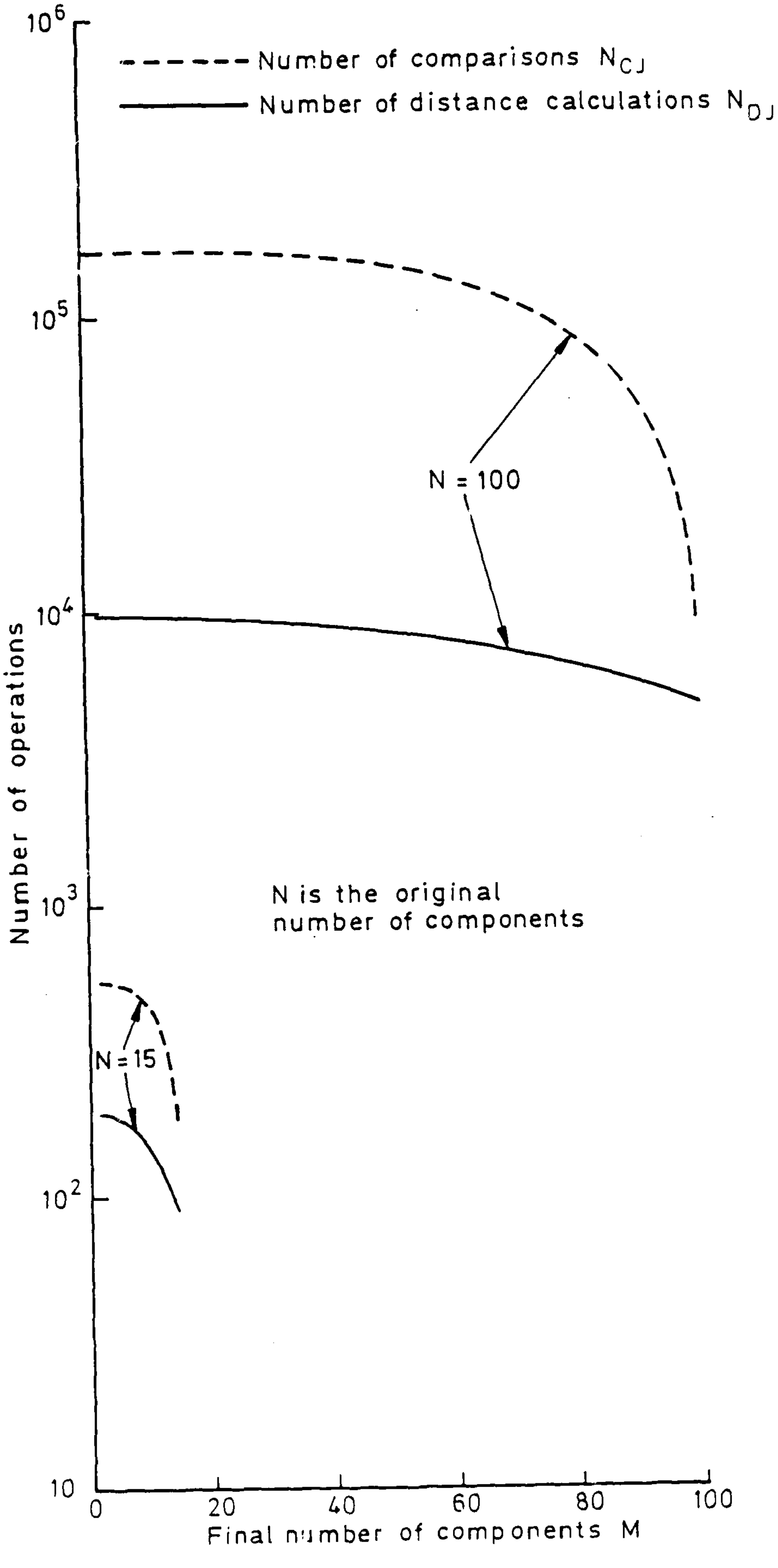
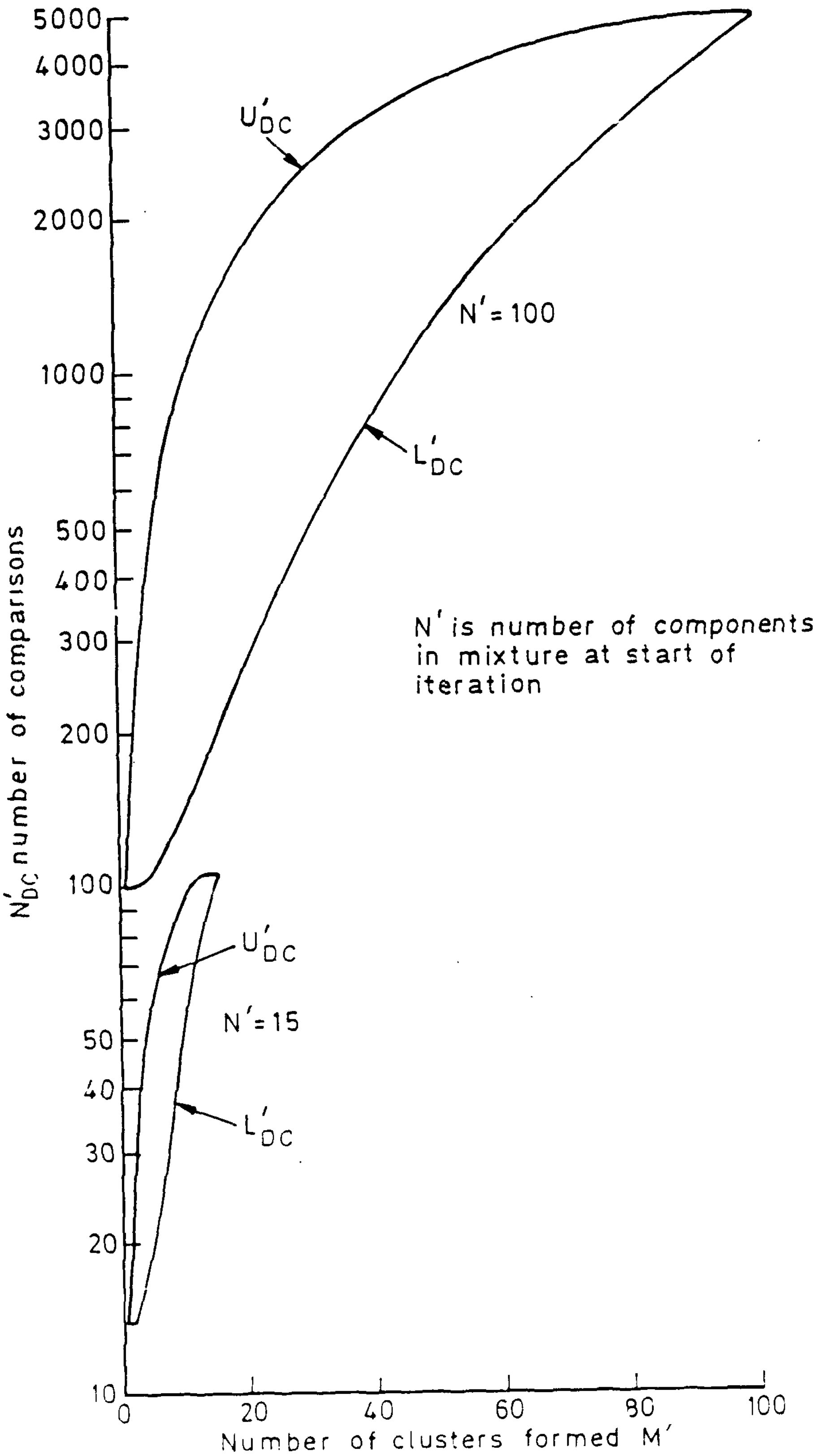
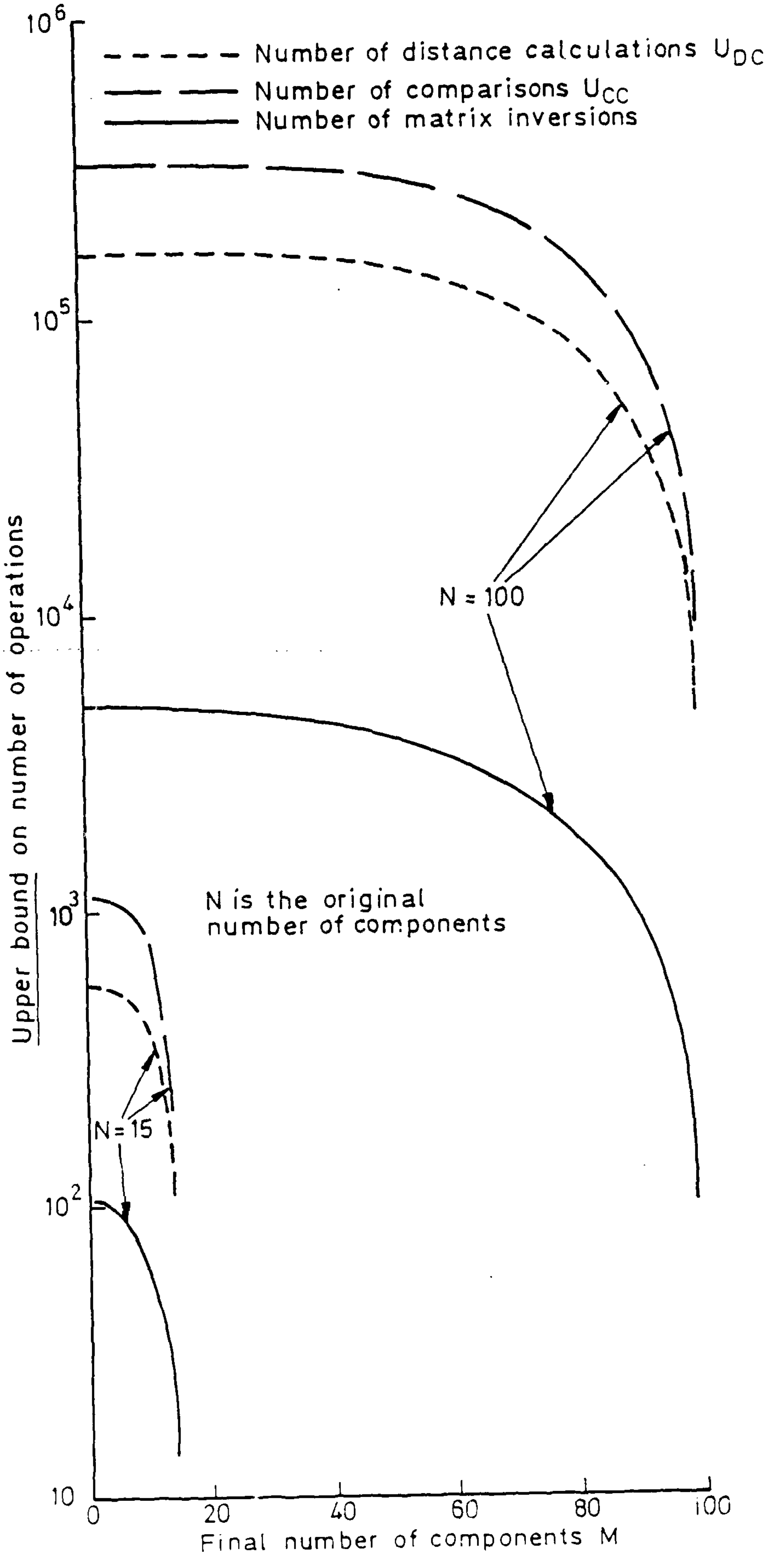


Fig D.1 Number of operations for the Joining Algorithm



[illegible] on the number of comparisons for a single iteration of the [illegible] algorithm



... bound on the number of operations for the Clustering Algorithm

Appendix E

RECURSIVE SOLUTION OF MULTIPLE SENSOR FILTER OF SECTION 7.3

Since only one measurement from each of the N_s sensors may be true, a measurement association hypothesis on the data at a particular time step may be denoted

$$\Psi_\ell = \left(\psi_{1j_i}, \psi_{2j_2}, \ldots, \psi_{uj_u}, \ldots, \psi_{N_s j_{N_s}}\right),$$

where ψ_{uj_u} indicates that the j_u^{th} measurement from sensor u is true if $j_u \neq 0$ and ψ_{uo} indicates that all measurements from sensor u are false. Since each sensor is independent, information from each sensor may be incorporated sequentially using the update relations of Chapter 2. Suppose that data from the first $u-1$ of the N_s sensors have been incorporated and let $\mathscr{H}^*_{u-1\ i}$ denote a hypothesis on the measurements from the $u-1$ sensors and from all previous time steps (the subscript k and the conditioning on $\mathscr{P}$ have been omitted). The subscript i, which enumerates all these hypotheses, runs from 1 to n^*_{u-1}. To incorporate measurements from sensor u, the set of feasible hypotheses must be widened to include

$$\left.\begin{aligned} &\mathscr{H}^*_{u\ell} = \left(\mathscr{H}^*_{u-1\ i}, \psi_{uj}\right) \\ \text{for}\quad & \\ &i = 1, \ldots, n^*_{u-1} \quad \text{and} \quad j = 0, \ldots, m_u \\ \text{and}\quad & \\ &\ell = (i-1)(m_u + 1) + j + 1 \end{aligned}\right\} . \qquad \text{..... (E-1)}$$

Thus the posterior pdf of $\underline{x}$ after the inclusion of measurements from sensor u is given by

$$p\left(\underline{x} \middle| Z_u^*\right) = \sum_{j=0}^{m_u} \sum_{i=1}^{n_{u-1}^*} p\left(\underline{x} \middle| \psi_{uj}, \mathscr{H}_{u-1\ i}^*, Z_u^*\right) \Pr\left\{\psi_{uj}, \mathscr{H}_{u-1\ i}^* \middle| Z_u^*\right\}$$

where $Z_u^* = \left\{ Z_1 , Z_2 , \ldots , Z_u \right\}$.

As in section 2.3 2, it can be shown that

$$\left.\begin{aligned}
& p\left(\underline{x} \middle| \psi_{uj}, \mathscr{H}_{u-1\ i}^*, Z_u^*\right) = \mathscr{N}\left(\underline{x} ; \hat{\underline{x}}_{uij}^*, P_{uij}^*\right) \\
& \text{where } \hat{\underline{x}}_{uij}^* = \bar{\underline{x}}_{ui}^* + \begin{cases} K_{ui}^*\left(\underline{z}_{uj} - H_u \bar{\underline{x}}_{ui}^*\right) & \text{if} \quad j \neq 0 \\ 0 & \text{if} \quad j = 0 \end{cases} \\
& K_{ui}^* = P_{uij}^* H_u^T R_u^{-1} \quad \text{for} \quad j_u \neq 0 \\
& P_{uij}^* = \begin{cases} M_{ui}^* - M_{ui}^* H_u^T S_{ui}^{*-1} H_u M_{ui}^* & \text{if} \quad j \neq 0 \\ M_{ui}^* \quad \text{if} \quad j = 0 \end{cases} \\
& \text{and} \\
& S_{ui}^* = H_u M_{ui}^* H_u^T + R_u
\end{aligned}\right\} . \tag{E-2}$$

In the above relations $\bar{\underline{x}}_{ui}^*$ and M_{ui}^* are the mean and covariance of the Gaussian distribution of $\underline{x}$ under hypothesis $\mathscr{H}_{u-1\ i}^*$ and so are available from the processing of data from sensor $u-1$. Note that with a minor change of notation, equation (E-2) is identical to equations (2.8) and (2.9). Likewise, by analogy with section 2.3.2,

$$\Pr\left\{\psi_{uj}, \mathscr{H}^*_{u-1\ i} \middle| Z^*_u\right\} = \begin{cases} \dfrac{\beta^*_{u-1\ i}\,\mathcal{N}\left(\underline{z}_{uj}\ ;\ H_u\underline{\bar{x}}^*_{ui}\ ,\ S^*_{ui}\right)}{E} & \text{for } j \neq 0 \\[2ex] \dfrac{\beta^*_{u-1\ i}\left(1 - P_{Du}\right)\rho_u}{EP_{Du}} & \text{for } j = 0 \end{cases}$$

..... (E-3)

where $\beta^*_{u-1\ i} = \Pr\left\{\mathscr{H}^*_{u-1\ i} \middle| Z^*_{u-1}, \mathscr{P}\right\}$, which is available from processing of sensor $u-1$, and E is the normalizing denominator. Again with a minor change of notation equation (E-3) is identical to equation (2.18), and so measurements from extra sensors can be processed using the same computer code as for the single sensor case.

After updating from sensor u is complete, quantities may be re-labelled ready for processing sensor $u+1$:

$$\underline{\bar{x}}^*_{u+1\ \ell} = \underline{\hat{x}}^*_{uij}$$

$$M^*_{u+1\ \ell} = P^*_{uij}$$

$$\beta^*_{u\ell} = \Pr\left\{\psi_{uj}, \mathscr{H}^*_{u-1\ i} \middle| Z^*_u, \mathscr{P}\right\}$$

$$n^*_u = n^*_{u-1}\left(m_u + 1\right)$$

and ℓ and $\mathscr{H}^*_{u\ell}$ are given by equation (E-1). When data from all the N_s sensors have been processed, the pdf of $\underline{x}$ is projected forwards to the next time step as described in section 2.3.3. Thus the solution is complete.

REFERENCES

No.	Author	Title, etc
1	S.S. Blackman	Multiple-target tracking with radar applications. Artech House (1986)
2	Y. Bar-Shalom T.E. Fortmann	Tracking and data association. Academic Press (1988)
3	Y. Bar-Shalom	Tracking methods in a multi-target environment. IEEE Trans on Automatic Control, Vol AC-23, No.4, August 1978
4	A. Farina S. Pardini	Survey of radar data-processing techniques in air-traffic-control and surveillance systems. IEE Proceedings, Vol 127, Part F, No.3, June 1980
5	I.R. Goodman H.L. Wiener W.W. Willman	Naval ocean-surveillance correlation handbook (1979) NRL Report 8402, Naval Research Laboratory, Washington, DC, September 1980
6	P. Smith G. Buechler	A branching algorithm for discriminating and tracking multiple objects. IEEE Trans on Automatic Control, Vol AC-20, No.1, February 1975
7	C.L. Morefield	Application of 0-1 integer programming to multitarget tracking problems. IEEE Trans on Automatic Control, Vol AC-22, No.3, June 1977
8	R.W. Sittler	An optimal data association problem in surveillance theory. IEEE Trans on Military Electronics, Vol MIL-8, No.2, April 1964
9	J.J. Stein S.S. Blackman	Generalized correlation of multi-target track data. IEEE Trans on Aerospace and Electronic Systems, Vol AES-11, No.6, November 1975

REFERENCES (continued)

No.	Author	Title, etc
10	R.A. Singer R.G. Sea K.B. Housewright	Derivation and evaluation of improved tracking filters for use in a dense multi-target environment. IEEE Trans on Information Theory, Vol IT-20, No.4, July 1974
11	Y. Bar-Shalom E. Tse	Tracking in a cluttered environment with probabilistic data association. Automatica, Vol 11, September 1975
12	K. Birmiwal Y. Bar-Shalom	On tracking a manoeuvring target in clutter. IEEE Trans on Aerospace and Electronic Systems, Vol AES-20, No.5, September 1984
13	Y. Bar-Shalom G.D. Marcus	Tracking with measurements of uncertain origin and random arrival times. IEEE Trans on Automatic Control, Vol AC-25, No.4, August 1980
14	A. Houlès Y. Bar-Shalom	Multisensor tracking of a manoeuvring target in clutter. Proceedings of NAECON (1987)
15	Y. Bar-Shalom K. Birmiwal	Consistency and robustness of PDAF for target tracking in cluttered environments. Automatica, Vol 19, No.4 (1983)
16	T.E. Fortmann Y. Bar-Shalom M. Scheffe S. Gelfand	Detection thresholds for tracking in clutter - a connection between estimation and signal processing. IEEE Trans on Automatic Control, Vol AC-30, No.3, March (1985)
17	Y. Bar-Shalom	Extension of the Probabilitistic Data Association filter to multi-target environment. Proceedings of the fifth Sumposium on Non-linear Estimation, San Diego, CA, September 1974

REFERENCES (continued)

No.	Author	Title, etc
18	T.E. Fortmann Y. Bar-Shalom M. Scheffe	Multi-target tracking using joint probabilistic data association. Proceedings of the 19th IEEE Conference on Decision and Control. Albuquerque, NM, December 1980
19	T.E. Fortmann Y. Bar-Shalom M. Scheffe	Sonar tracking of multiple targets using joint probabilistic data association. IEEE Journal of Oceanic Engineering, Vol OE-8, No.3, July 1983
20	K-C. Chang Y. Bar-Shalom	Joing probabilistic data association for multi-target tracking with possibly unresolved measurements and manoeuvres. IEEE Trans on Automatic Control, Vol AC-29, No.7, July 1984
21	K-C. Chang C-Y. Chong Y. Bar-Shalom	Joint probabilistic data association in distributed sensor networks. IEEE Trans on Automatic Control, Vol AC-31, No.10, October 1986
22	K-C. Chang Y. Bar-Shalom	A simplification of the JPDAM. IEEE Trans on Automatic Control, Vol AC-31, No.10, October 1986
23	D.B. Reid	An algorithm for tracking multiple targets. IEEE Trans on Automatic Control, Vol AC-24, No.6, December 1979
24	R.J. Kenefic	Optimum tracking of a manoeuvring target in clutter. IEEE Trans on Automatic Control, Vol AC-26, No.3, June 1981
25	D.L. Alspach	A Guassian sum approach to the multi-target identification-tracking problem. Automatica, Vol 11, No.3, May 1975

REFERENCES (continued)

No.	Author	Title, etc
26	S. Mori C-Y. Chong E. Tse R.P. Wishner	Tracking and classifying multiple targets without a priori identification. IEEE Trans on Automatic Control, Vol AC-31, No.5, May 1986
27	A.H. Jazwinski	Stochastic processes and filtering theory. Academic Press (1970)
28	R.C. Lee	Optimal estimation, identification and control. MIT Research Monograph 28 (1964)
29	A.E. Bryson Y.C. Ho	Applied optimal control. John Wiley and Sons (1975)
30	A.P. Sage J.L. Melsa	Estimation theory with applications to communications and control. Mc.Graw-Hill (1971)
31	J.S. Meditch	Stochastic linear estimation and control. Mc.Graw-Hill (1969)
32	A. Gelb	Applied optimal estimation. MIT Press (1974)
33	B.S. Everitt D.J. Hand	Finite mixture distributions. Chapman and Hall (1981)
34	D.M. Titterington A.F.M. Smith U.E. Makov	Statistical analysis of finite mixture distributions. John Wiley and Sons (1985)
35	V. Nagarajan R.N. Sharma M.R. Chidambara	An algorithm for tracking a manoeuvring target in clutter. IEEE Trans on Aerospace and Electronic Systems, Vol AES-20, No.5, September 1984
36	H.W. Sorenson	Parameter estimation, principles and problems. Marcel Dekker Inc. (1980)

REFERENCES (continued)

No.	Author	Title, etc
37	M. Athans R.H. Whiting M. Gruber	A sub-optimal estimation alorithm with probabilistic editing for false measurements with applications to target tracking with wake phenomena. IEEE Trans on Automatic Control, Vol AC-22, No.3, June 1977
38	K.R. Pattipati N.R. Sandell	A unified view of state estimation in switching environments. Proceedings of American Control Conference (1983)
39	A.G. Jaffer S.C. Gupta	On estimation of discrete processes under multiplicative and additive noise conditions. Information Sciences, Vol 3, No.3, July 1971
40	H.A.P. Blom	Overlooked potential for systems with Markovian coefficients. Proc. of 25th IEEE Conference on Decision and Control, Athens, December 1986
41	G.A. Ackerson K.S. Fu	On state estimation in switching environments. IEEE Trans on Automatic Control, Vol AC-15, No.1, February 1970
42	J.K. Tugnait	Detection and estimation for abruptly changing systems. Automatica, Vol 18, No.5, September 1982
43	J.K. Tugnait A.H. Haddad	A detection-estimation scheme for state estimation in switching environments. Automatica, Vol 15, No.4, July 1979
44	D.G. Lainiotis S.K. Park	On joint detection, estimation and system identification: discrete data case. International Journal of Control, Vol 17, No.3 (1973)
45	J.L. Weiss T.N. Upadhyay R. Tenney	Finite computable filters for linear systems subject to time varying model uncertainty. Proceedings of NAECON (1983)

REFERENCES (concluded)

No.	Author	Title, etc
46	T. Kailath	The divergence and Bhattacharyya distance measures in signal selection. IEEE Trans on Communication Technology, Vol COM-15, No.1, February 1967
47	D.J. Hand	Discrimination and classification. John Wiley (1981)
48	M.R. Anderberg	Cluster analysis for applications. Academic Press (1973)
49	A.W. Bridgewater	Analysis of second and third order steady-state tracking filters. From: AGARD Conference Proceedings No.252, Monterey, California (1978)
50	D.J. Salmond	The characteristics of the second-order target model. Royal Aircraft Establishment, Technical Report TR 85071, August 1985
51	M. Gauvrit	Bayesian adaptive filter for tracking with measurements of uncertain origin. Automatica, Vol 20, No.2 (1984)
52	H.A.P. Blom	An efficient filter for abruptly changing systems. Proc of 23rd IEEE Conference on Decision and Control, Las Vegas (1984)
53	D.J. Salmond	The Kalman tracking filter, the α-β filter and smoothing filters. Royal Aircraft Establishment, Technical Memorandum AW48 (1981)
54	D.J. Salmond	Fixed point smoothing filters and trajectory estimation. Royal Aircraft Establishment, Technical Memorandum AW58 (1982)
55	A.K. Mahalanabis Bin Zhou	A Joint Probabilistic Data Association Smoothing Algorithm for Multi-target Tracking. Proceedings of American Control Conference (1988)

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNLIMITED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification/Marking
	RAE TM AW 121		UNLIMITED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
7673000W	Royal Aerospace Establishment, Farnborough, Hants, UK
5a. Sponsoring Agency's Code	**6a. Sponsoring Agency (Contract Authority) Name and Location**

7. Title

Tracking in uncertain environments

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1. Surname, Initials	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Salmond, D.J.			September 1989	286	55

11. Contract Number	12. Period	13. Project	14. Other Reference Nos.

15. Distribution statement

(a) Controlled by –

(b) Special limitations (if any) –

If it is intended that a copy of this document shall be released overseas refer to RAE Leaflet No.3 to Supplement 6 of MOD Manual 4.

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)

Tracking filters. Mixture distributions. Bayesian filters. Kalman filters. Data fusion. PDAF.

17. Abstract

This study concerns the problem of tracking a target when the origin of sensor measurements is uncertain. The full Bayesian solution to this type of problem gives rise to Gaussian mixture distributions, which are composed of an ever increasing number of components. Two algorithms have been developed for approximating such distributions, so allowing practical tracking filters to be derived.

For a standard tracking problem, simulation has been used to determine the significant range of problem parameters where, at the expense of extra computation, the new algorithms give a substantial performance improvement over the well-known Probabilistic Data Association Filter. The algorithms have also been used to derive Bayesian filters for data fusion problems and for tracking a target in the presence of intermittent interfering measurements.

F5910/1